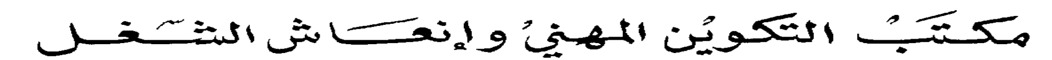


ROYAUME DU MAROC



Office de la Formation Professionnelle et de la Promotion du Travail

DIRECTION RECHERCHE ET INGÉNIERIE DE FORMATION

RÉSUMÉ THÉORIQUE & GUIDE DE TRAVAUX PRATIQUES

MODULE : 24 UTILISATION DE L'AUTOMATE PROGRAMMABLE

SECTEUR : ÉLECTROTECHNIQUE

SPÉCIALITÉ : ÉLECTROMÉCANIQUE DES SYSTÈMES AUTOMATISÉS

NIVEAU : TECHNICIEN SPÉCIALISÉ

ANNEE 2003

Document élaboré par :

Nom et prénom	EFP	DR
ELKORNO NAIMA	CFF / ISIC	

Révision linguistique
-
-
-

Validation
-
-
-

SOMMAIRE

MODULE 24 : UTILISATION DE L'AUTOMATE PROGRAMMABLE

Code : **Durée : 75 h**

OBJECTIF OPÉRATIONNEL DE PREMIER NIVEAU
DE COMPORTEMENT

COMPORTEMENT ATTENDU

Pour démontrer sa compétence le stagiaire doit
utiliser un automate programmable
selon les conditions, les critères et les précisions qui suivent.

CONDITIONS D'ÉVALUATION

- À partir de directives.
- À l'aide :
  - de fiches techniques et du manuel d'utilisation d'un automate;
  - d'un programme en langage Grafcet ou en diagramme à échelons;
- Sur un automate programmable avec E/S " tout ou rien ".

CRITÈRES GÉNÉRAUX DE PERFORMANCE

- Utilisation appropriée de l'équipement informatique.
- Respect des méthodes et des conventions de programmation d'un automate.
- Respect des normes.

(à suivre)

OBJECTIF OPÉRATIONNEL DE PREMIER NIVEAU DE COMPORTEMENT(suite)

PRÉCISIONS SUR LE COMPORTEMENT ATTENDU	CRITÈRES PARTICULIERS DE PERFORMANCE
A. Raccorder un automate.	- Repérage de l'information pertinence dans la documentation technique. - Localisation précise des points de raccordement - Câblage conforme au schéma de raccordement. - Reconnaissance précise des composants associés au matériel et des composants associés au logiciel.
B. Accéder aux fonctions d'un automate.	- Configuration précise de l'automate. - Utilisation appropriée du logiciel ou de la console de programmation dédiée. - Détermination juste du mode d'adressage.
C. Déceler des problèmes de fonctionnement d'un automatisme simple commandé par un automate.	- Respect de la procédure d'interrogation des E/S et des cases mémoire. - Indication juste des problèmes de fonctionnement.
D. Apporter des modifications mineures au programme d'un automate	- Modification conforme à la demande. - Respect de la procédure d'éducation. - Programmation précise des ajouts ou des retraits. - Respect de la procédure de sauvegarde.
E. Effectuer l'essai d'un automatisme simple commandé par un automate.	- Programmation fonctionnelle en simulation. - Système fonctionnel à la suite de la modification.

OBJECTIFS OPÉRATIONNELS DE SECOND NIVEAU

LE STAGIAIRE DOIT MAÎTRISER LES SAVOIRS, SAVOIR-FAIRE, SAVOIR PERCEVOIR OU SAVOIR ÊTRE JUGÉS PRÉALABLES AUX APPRENTISSAGES DIRECTEMENT REQUIS POUR L'ATTEINTE DE L'OBJECTIF DE PREMIER NIVEAU, TELS QUE :

Avant d'apprendre à raccorder un automate (A) :

1. Décrire l'architecture d'un automate.
2. Décrire les applications d'un automate.
3. Reconnaître les différents modules E/S.
4. Repérer un logiciel de programmation.

Avant d'apprendre à accéder aux fonctions d'un automate (B) :

5. Distinguer les langages de programmation d'un automate.
6. Décrire les principales instructions d'un automate.
7. Utiliser un logiciel de programmation.

Avant d'apprendre à déceler des problèmes de fonctionnement d'un automatisme simple commandé par un automate (C) :

8. Reconnaître l'état logique des E/S dans un programme par le mode dynamique (en ligne).

Avant d'apprendre à effectuer l'essai d'un automatisme simple commandé par un automate (E) :

9. Reconnaître les dangers potentiels.

PRÉSENTATION DU MODULE

Ce module de compétence générale est enseigné au troisième semestre du programme. Il est préalable aux apprentissages réalisés au module 29 (Système automatisé contrôlé par A P I) et nécessite comme pré requis les modules 18 (Logique combinatoire) et 19 (Logique séquentielle).

L'objectif de ce module est de faire acquérir les connaissances relatives à l'architecture d'un automate ,aux divers modes d'entrées et de sorties, aux langages de programmation, au mode de communication, ainsi qu'à la vérification de l'état logique des éléments en vue d'un diagnostic. Il vise donc à rendre le stagiaire apte à utiliser un automate programmable

La durée totale de ce module est de 75 H dont la théorie représente 38 % et la pratique 62% L'évaluation incluse.

Module24 UTILISATION DE L'AUTOMATE PROGRAMMABLE
RESUME THEORIQUE

I - Raccordement d'un automate programmable

I.1 - Introduction

Les automatismes sont réalisés en vue d'apporter des solutions à des problèmes de nature technique, économique ou humaine.

Eliminer les tâches dangereuses et pénibles, en faisant exécuter par la machine les tâches humaines complexes ou indésirables.

Améliorer la productivité en asservissant la machine à des critères de production, de rendement ou de qualité.

Piloter une production variable, en facilitant le passage d'une production à une autre.

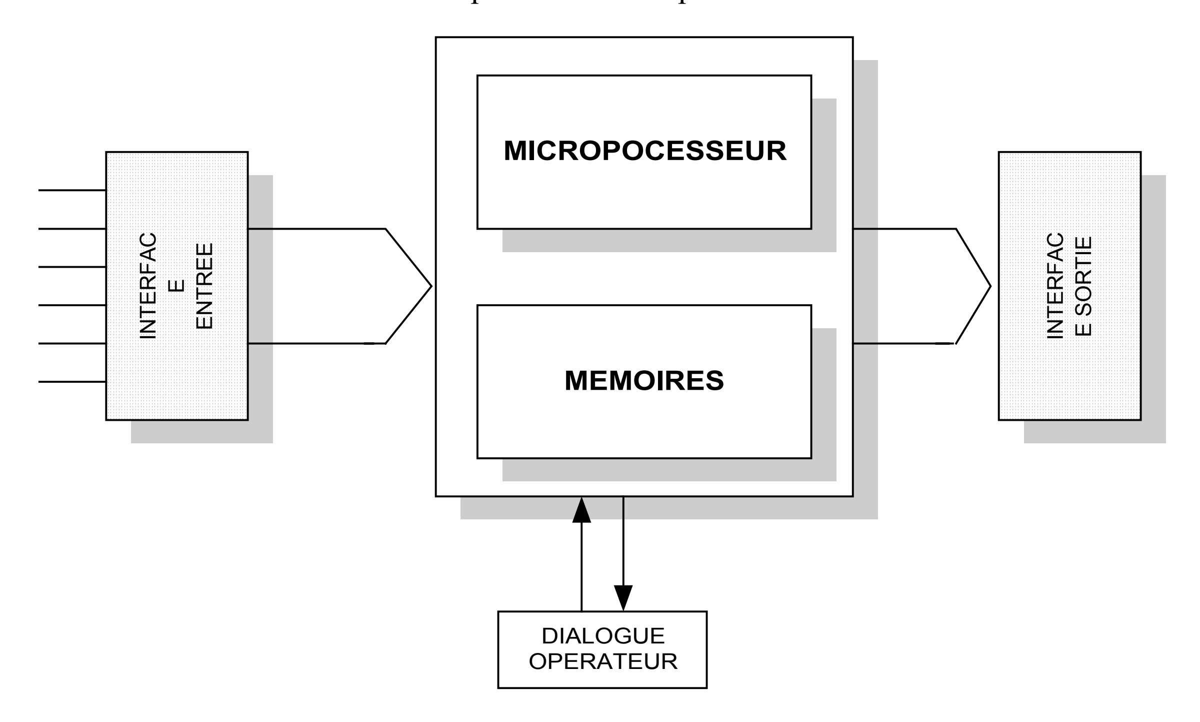
Renforcer la sécurité en surveillant et contrôlant les installations et machines.

On distingue dans tout système automatisé la machine ou l'installation et la partie commande constituée par l'appareillage d'automatisme. Cette partie commande est assurée par des constituants répondant schématiquement à quatre fonctions de base :

- L'acquisition des données
- Le traitement des données
- La commande de puissance
- Le dialogue homme machine

I.2 - Structure fonctionnelle de l'automate :

L'automate programmable industriel est un appareil qui traite les informations selon un programme préétabli.
Son fonctionnement est basé sur l'emploi d'un micro processeur et de mémoires.



Structure d'un système de traitement

I.2.1 - Interface d'entrée

Elles permettent d'isoler électroniquement le circuit externe (saisie de l'information) du circuit de traitement.

I.2.2 - L' unité centrale:

cœur de l'automate, elle est constituée:

- d'un **processeur** qui exécute le programme
- de **mémoires** qui, non seulement contiennent ce programme, mais aussi des informations de données (durée d'une temporisation, contenu d'un compteur)

Les types de mémoires :

Mémoires vives:

RAM – Random Access Memory (Mémoire à accès aléatoires)
Ce sont des mémoires volatiles.Lues et écrites par le processeur.

Mémoires mortes:
ROM – Read only memory
PROM – ROM programmable
NE PEUVENT PAS ETRE EFFACES
REPROM – effacement par UV
EEPROM – effacement électrique

I.2.3 - Interface de sortie

Elles permettent de commander les sorties toute ou rien (TOR) telle que : les contacteurs, les moteurs pas à pas, les électrovannes et ainsi des sorties analogiques (boucle de régulation débit température et variateur de vitesse.)

I.2.4 - Communication et dialogue

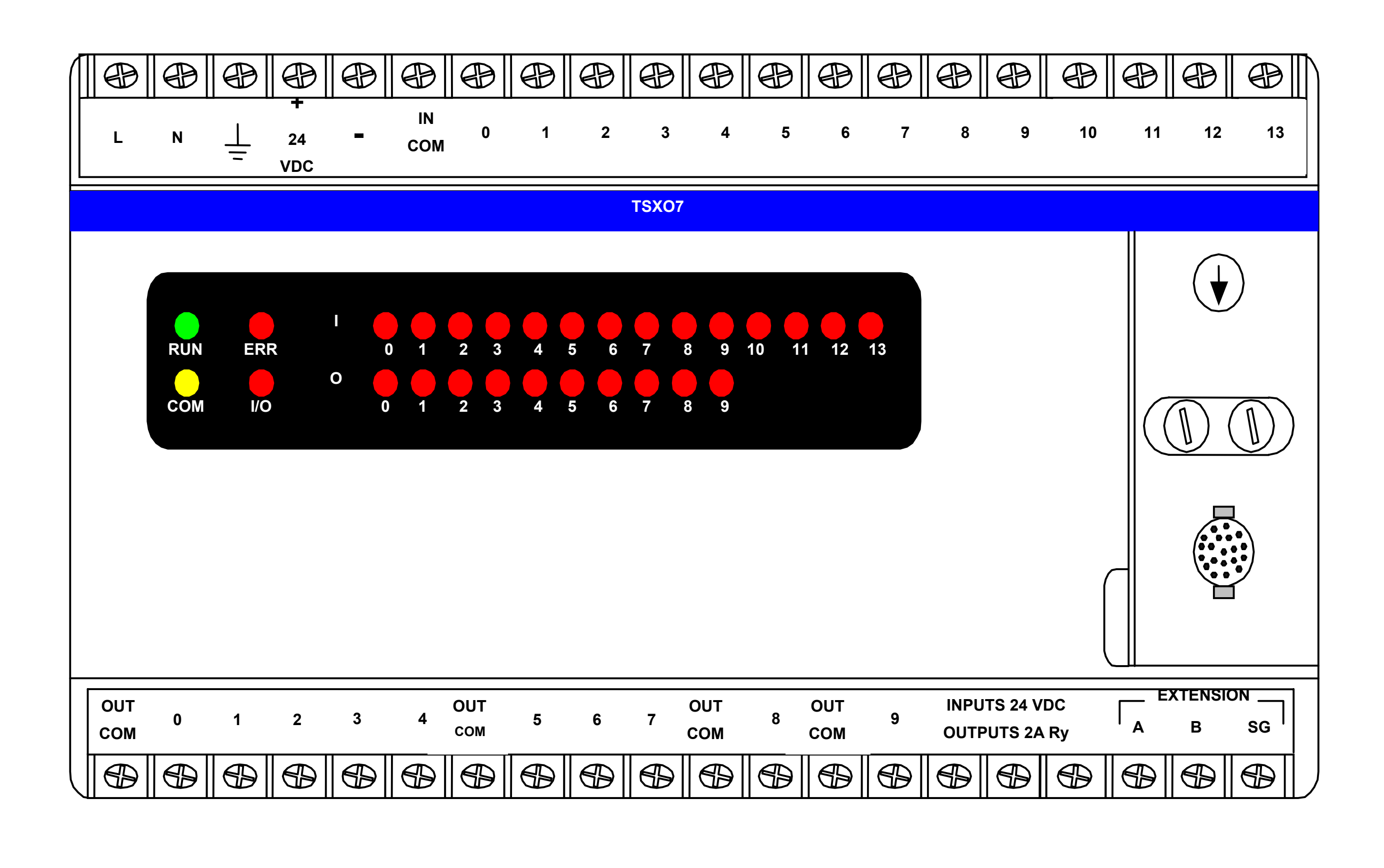
Elle est réalisée avec l'opérateur par un pupitre de dialogue ou par l'intermédiaire d'un ordinateur et avec les autres automates pour un réseau informatique local.

I.3 - Description des automates :

Il existe deux types d'automate programmable industriel:
- le type **monobloc**
- le type **modulaire**

I.3.1 - Automate Monobloc :

Le type **monobloc** possède généralement un nombre d'entrées et de sorties restreint et son jeu d'instructions ne peut être augmenté. Bien qu'il soit parfois possible d'ajouter des extensions d'entrées/sorties, le type monobloc a pour fonction de résoudre des automatismes simples faisant appel à une logique séquentielle et utilisant des informations tout-ou-rien.



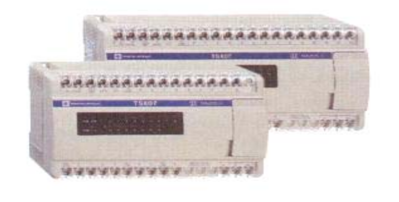




FIGURE 1 :Automate monobloc TSX Nano

http://module01-ofppt.blogspot.com/

Exemple 1 : automate monobloc

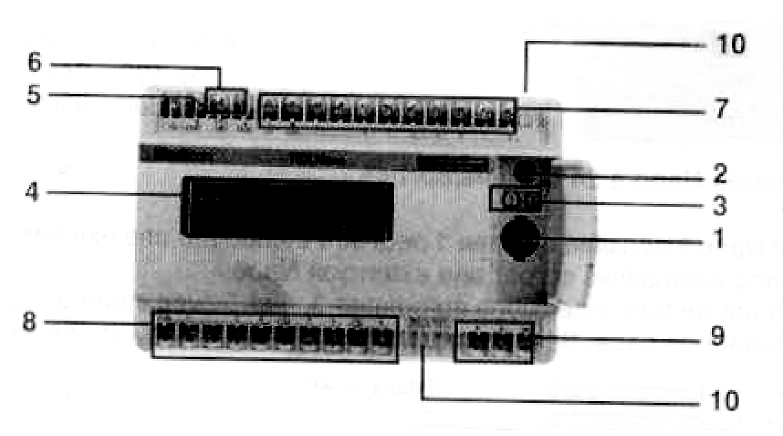


Fig. 2 automate monobloc

1- Une prise (1) pour raccordement du terminal de programmation .
2- Un sélecteur pour codage de la fonction base / extension.
3- Deux points de réglage analogique.
4- Une visualisation :
- Des entrées 0 à 8 ou 0 à 13 et sorties 0 à 6 ou 0 à 9,
- De l'état automate (RUN, ERR, COM, I/O).
5- Un raccordement de l'alimentation secteur
6- Une alimentation capteurs (=24V/150mA) sur modèles alimentés en ~ 100…240V.
7- Un raccordement des capteurs d'entrées.
8- Un raccordement des préactionneurs de sorties.
9- Un raccordement extension (extension d'entrées /sorties et / ou extension automate) ou raccordement Modbus esclave
10- Un cache amovible pour protection des borniers à vis.

I.3.2 - Automate Modulaire:

Par ailleurs, le type **modulaire** est adaptable à toutes situations. Selon le besoin, des modules d'entrées/sorties analogiques sont disponibles en plus de modules spécialisés tels: PID, BASIC et Langage C, etc. La modularité des API permet un dépannage rapide et une plus grande flexibilité. La figure 3 présente un automate modulaire.

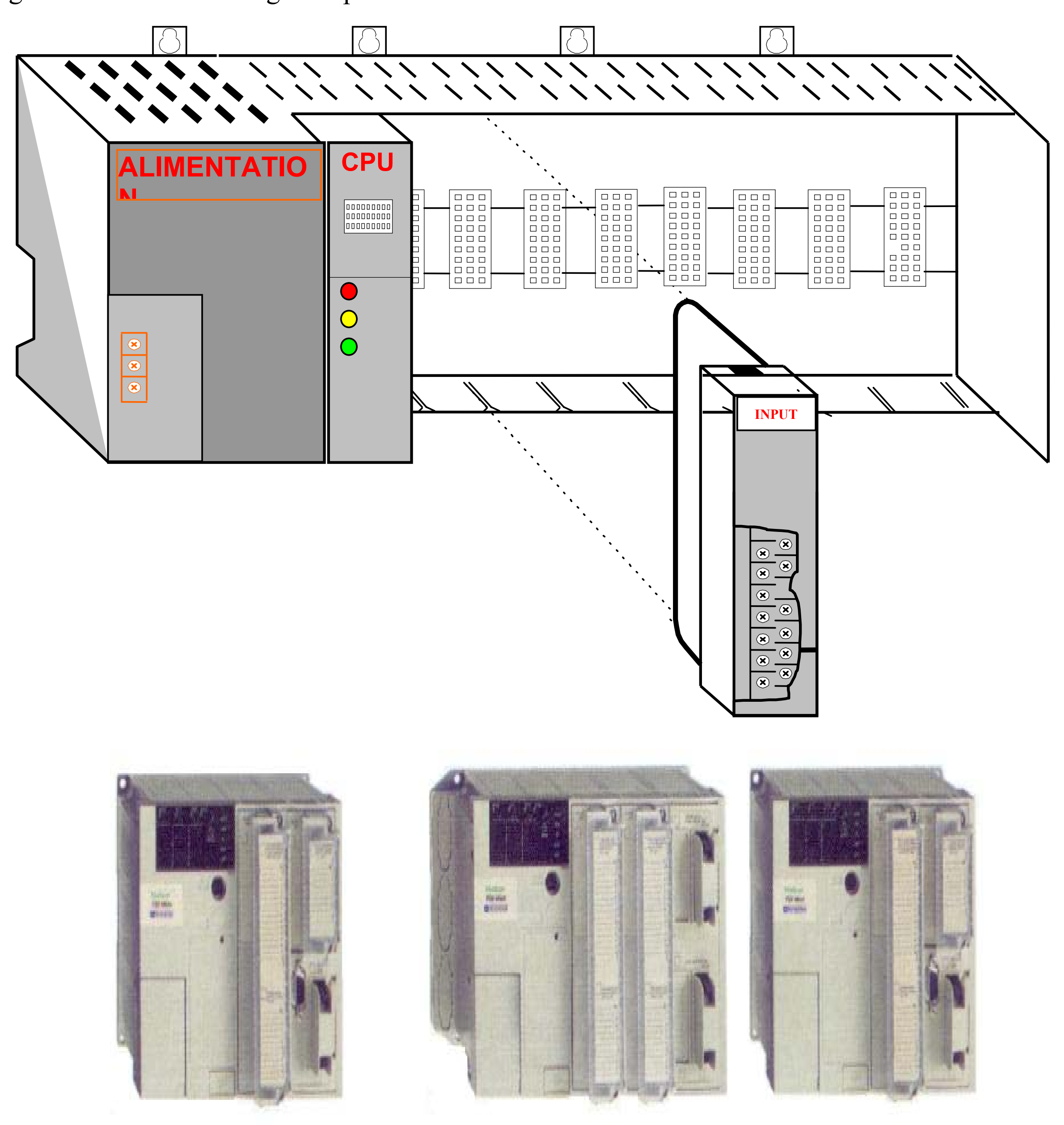


FIGURE 3:API MODULAIRE

Exemple 2 :automate modulaire :

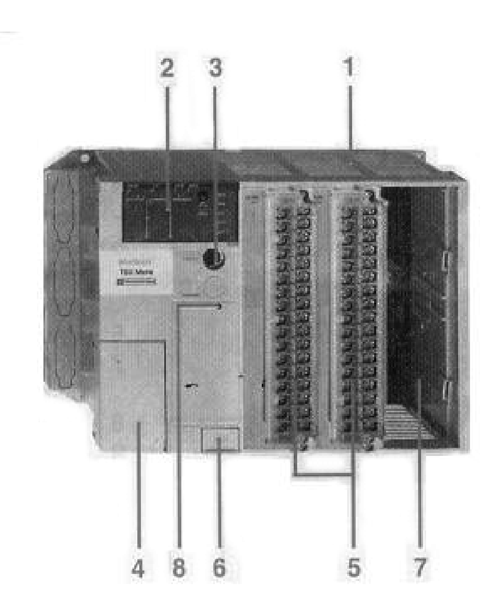


Fig. 4 automate modulaire :

L'automate TSX 37-08 comprend :

1- Un bac à 3 emplacement.
2- Un bloc de visualisation centralisé.
3- Une prise terminal repérée TER.
4- Une trappe d'accès aux bornes d'alimentation.
5- Deux modules à 16 entrées et 12 sorties « Tout ou Rien » positionnés dans le premier et le deuxième emplacements (positions 1, 2, 3 et 4).
6- Une trappe d'accès à la pile optionnelle.
7- Un emplacement disponible.
8- Un bouton de réinitialisation

I.4 - Les applications de l'automate :

Les automates trouvent leur application en milieu industriel, domestiques. On cite quelques exemples courants :

Exemple n°1: Feux de carrefour

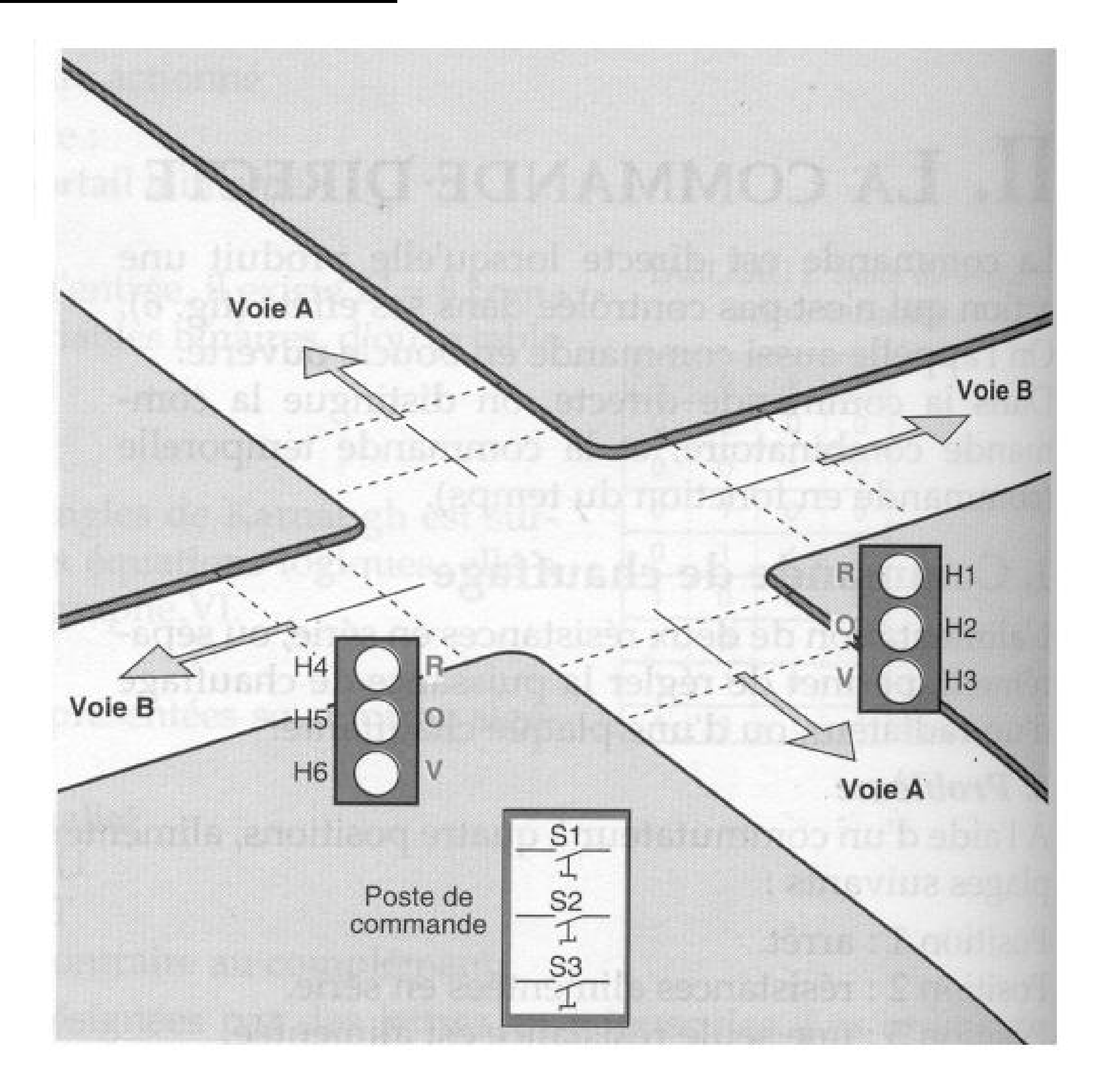


Fig. 5 : Feux de carrefour

Description

On régle la circulation d'un carrefour de deux voies A et B par des feux tricolores (Rouges, orange, vert).

Exemple 2 : Portail coulissant.

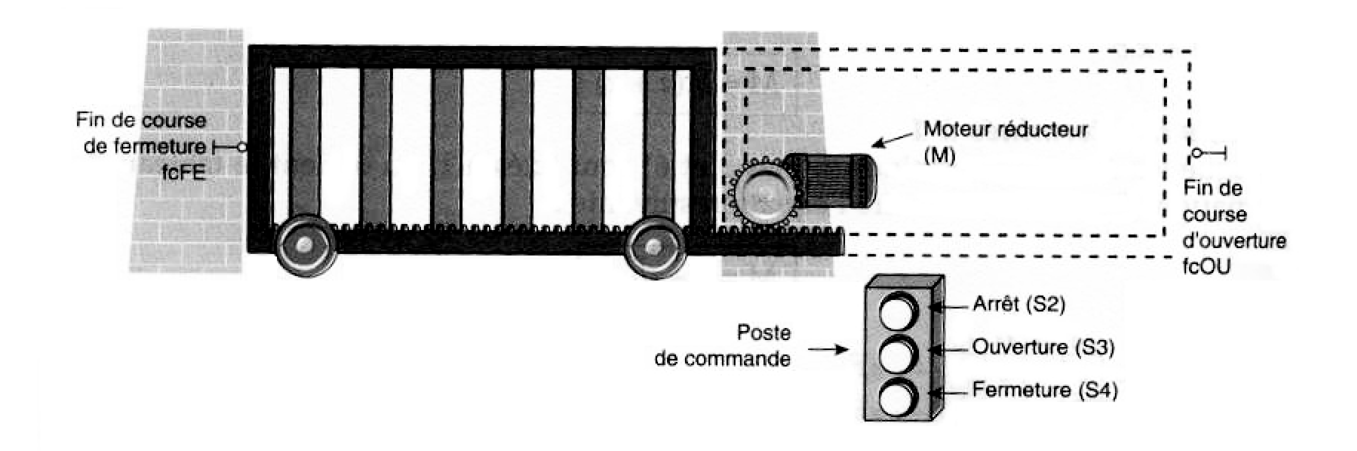


Fig. 6 : Portail coulissant.

Soit un portail coulissant à commander :

- Le portail étant fermé, le contact fin de course fcFE est actionné ;
- On appuie sur le bouton-poussoir d'ouverture S3, le moteur actionne le portail et provoque son ouverture ;
- En fin d'ouverture, le contact fin de course fcOU est actionné, il signale l'ouverture du portail, et il coupe l'alimentation du moteur.

L'action sur le bouton-poussoir de fermeture provoque l'inversion de sens de marche du moteur, et la fermeture du portail.

Le portail libère le contact fcOU, et se déplace jusqu'à actionner le contact fcFE qui provoque l'arrêt du moteur.

Exemple 3 système de perçage

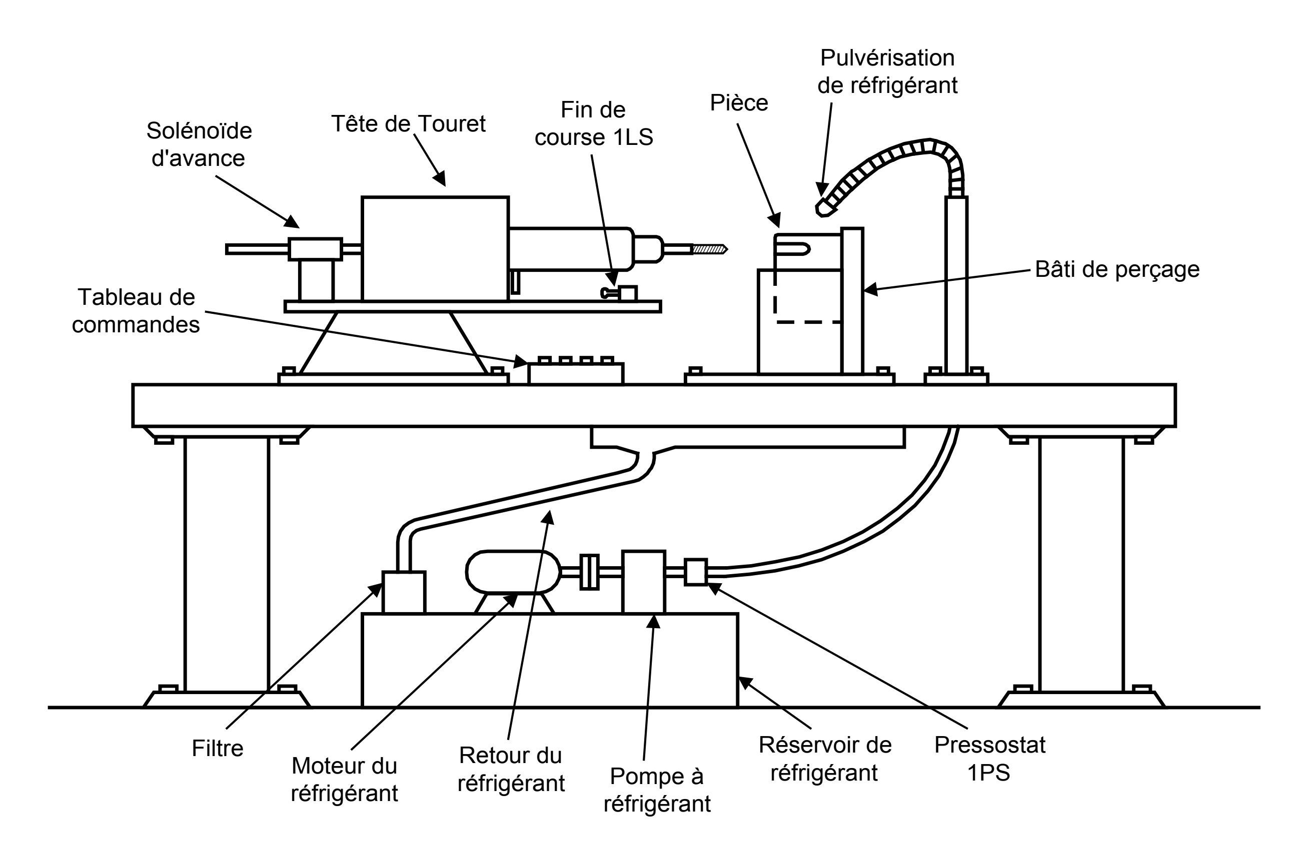


Fig. 7 système de perçage

Exemple 4° système de pompage

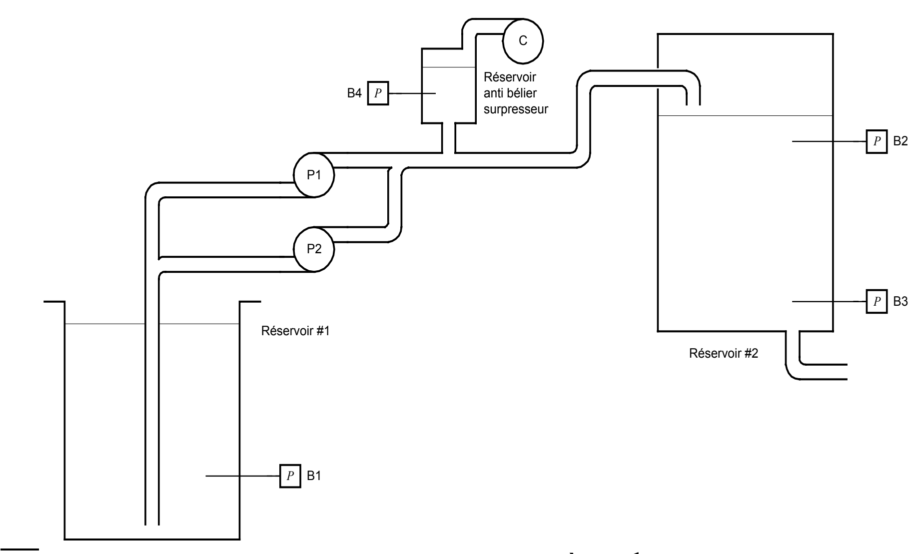


Fig. 8 système de pompage

I.5 - Les différents modules d'entrée/ sortie :

Les modules d'entrée / sortie sont les interfaces qui permettent de communiquer avec le micro presseur .
On distingue :
- Les interfaces d'entrée.
- Les interface de sortie.

I.5.1 - Interface d'entrée :

a) Interface Tout on rien (TOR)

A partir d'un signal quelconque en entrée, les interfaces fournissent en sortie deux tensions 0V ou 5V Ces interfaces sont de type à contact, ou statique.

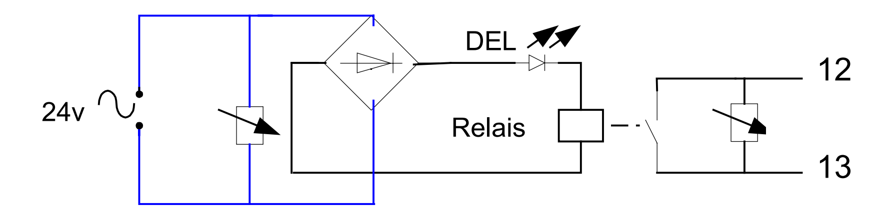


Fig. 9 : Interface d'entrée tout ou rien à relais

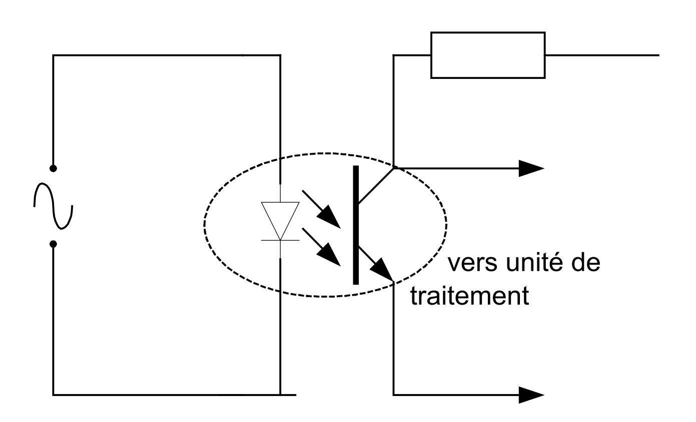


Fig. 10 : Interface d'entrée tout ou rien statique.

b) Transmetteurs analogiques

Les transmetteurs analogiques : Tension / intensité permettent d'adapter les signaux issus des capteurs pour les rendre compatibles avec l'unité de traitement. La variation de la grandeur d'entre est convertie en une variation :

- En tension : de 0V, à10V
- En intensité : de 0 Ma à 20mA, ou de 4 mA à 20mA

Exemple : Transmission de mesure de température effectuée par une sonde PT

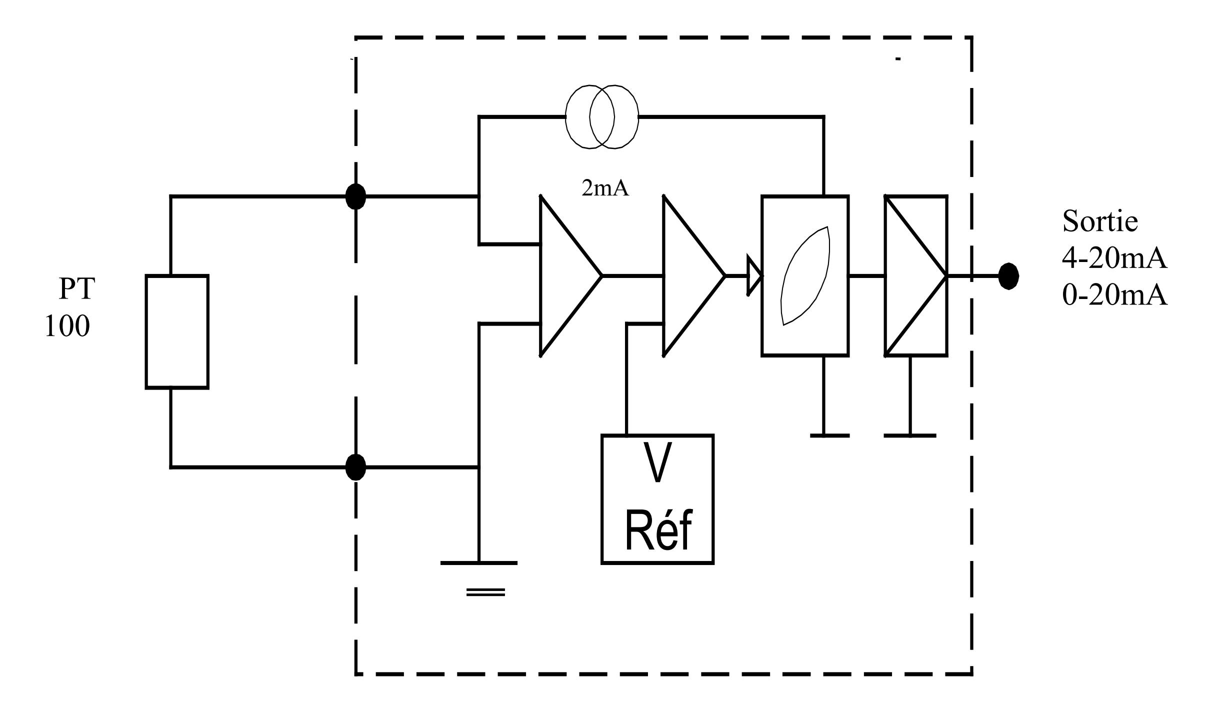


Fig. 11 Interface d'entrée analogique

I.5.2 - Interface de sortie :

a) Interface de sortie tout ou rien

La sortie de l'unité de traitement peut s'effectuer soit sur relais, soit sur transistor (TTL), ou avec un triac.

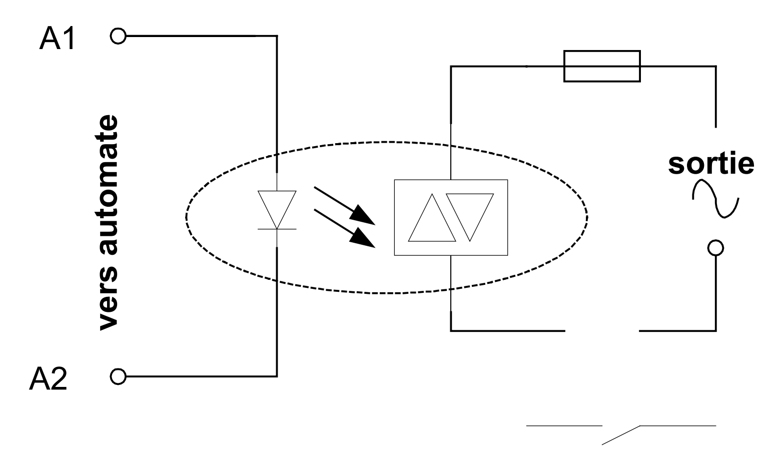


fig.12 Interface de sortie statique avec triac

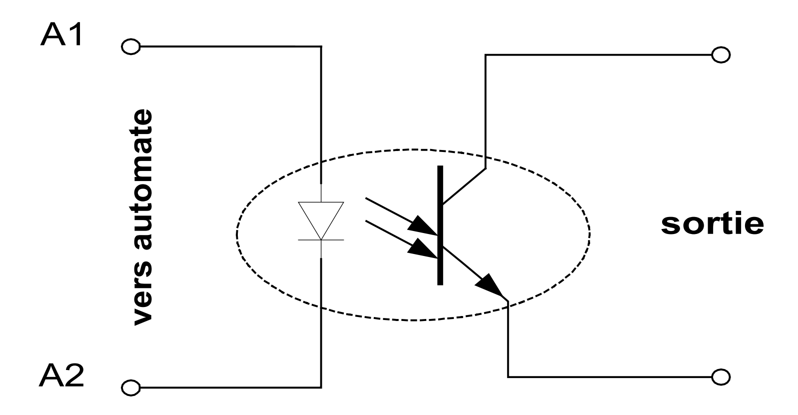


fig.13 Interface de sortie statique à transistor

b) Interfaces de sorties analogiques

Les conventions digitaux /analogiques ont pour fonction de générer un signal analogique normalisé (0-10 V ;0-20mA) à partir d' une information numérique, délivrée par l' unité de traitement et codée en binaire, sur des sorties digitales TOR raccordées aux entrées de l' interface(ou convertisseur).

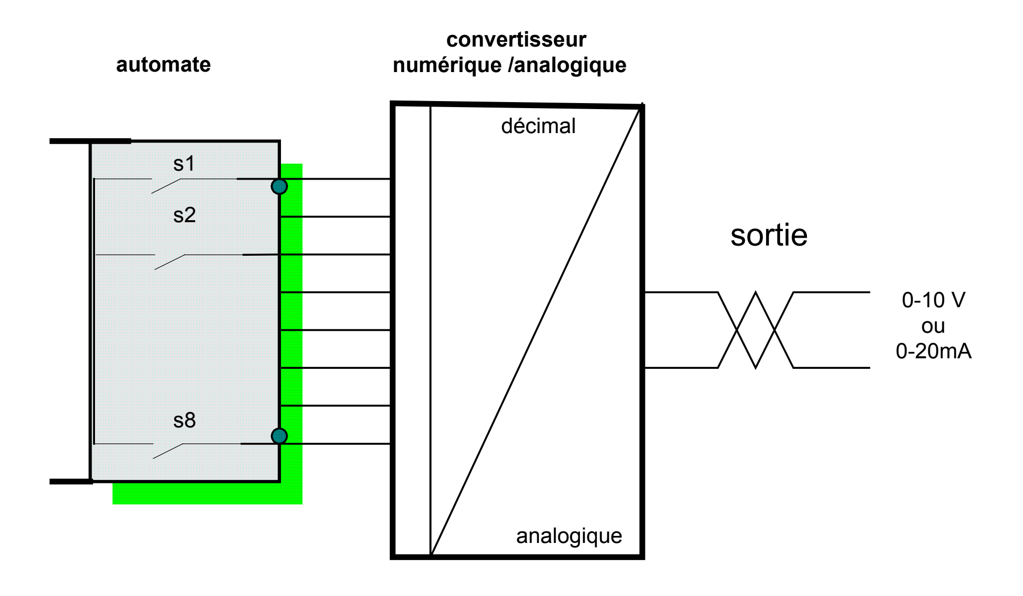


fig14 Interface de sortie numérique/analogique

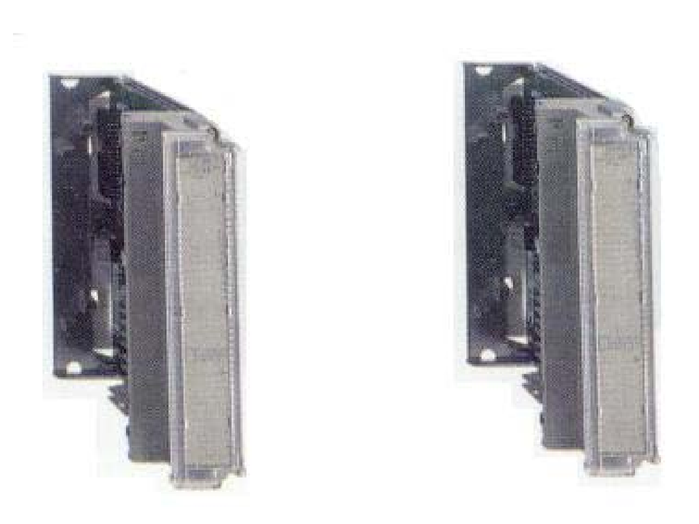


Fig. 15 Modules E/S



Fig 16 Les bornes sont des E/S

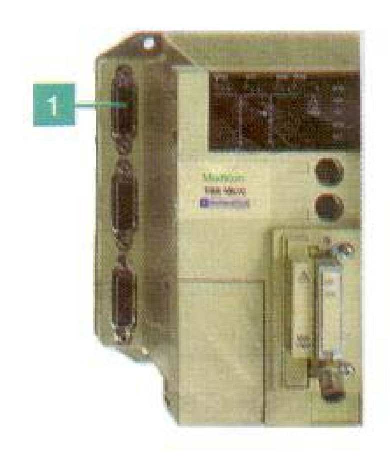
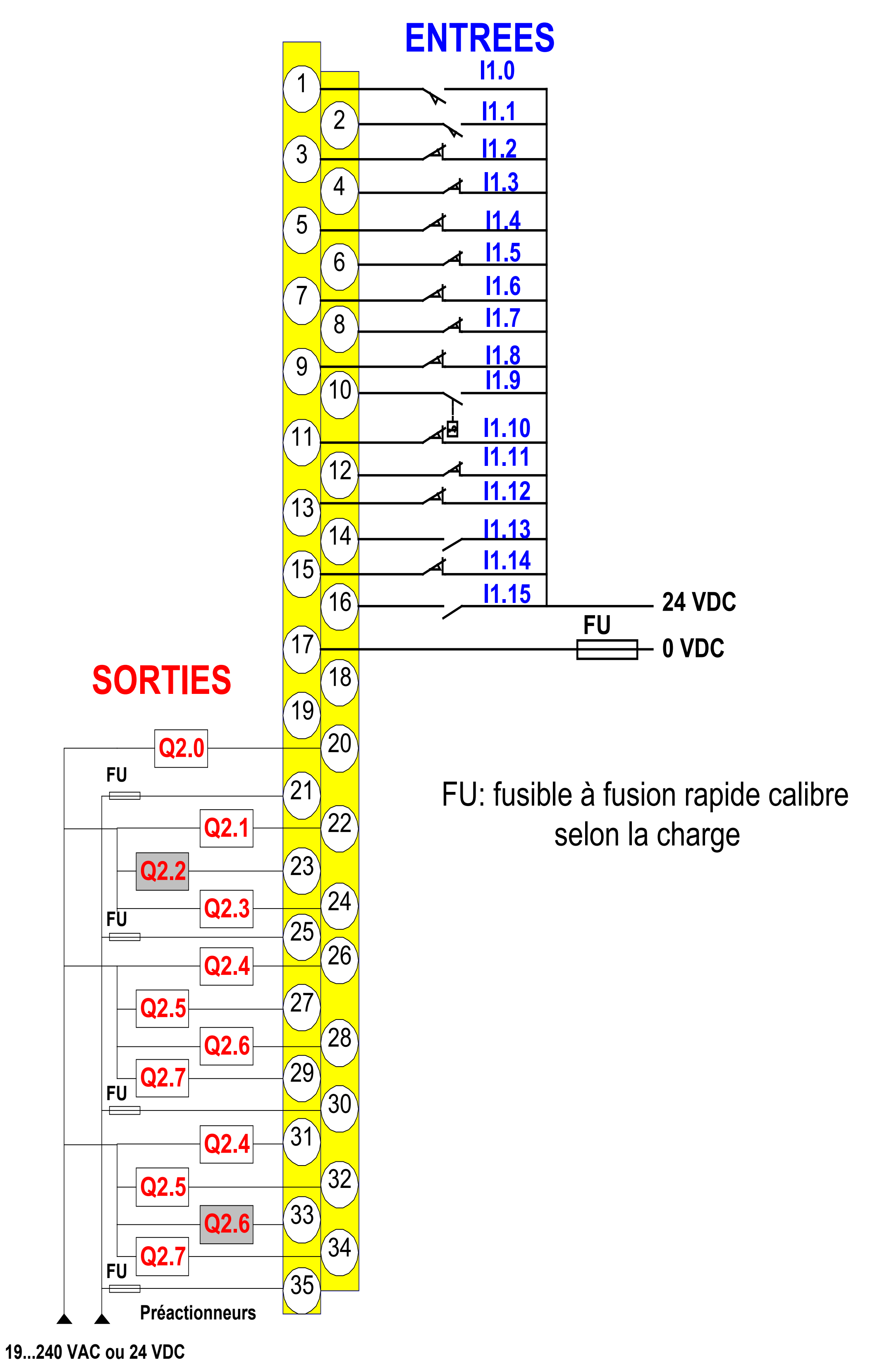


Fig 17 : Le n° 1 est un connecteur pour une entrée sortie analogique



F IG 18 Câblage des entrées/sorties TOR :

I.6 - Les logiciels de programmation :

Marque	Automate	Logiciel
Télemecanique	TSX Nano	Pl707
	TSX 3708,TSx22	Pl7-micro
	TSX Premium	Pl7 junior
ALENBRADLEY	SLC 500	APSF
SIEMENS	Serie5:S5 Serie7:S7	Step 5 Step 7

Ce tableau récapitulatif donne le logiciel et le type d'automate conforme à ce dernier .L'opérateur peut communiquer avec l'automate soit à travers un P.C portable ,fixe (Fig.18) ou avec la console (Fig. 19,20). On lie l'automate au PC (ou à la console) par un câble(RS232)

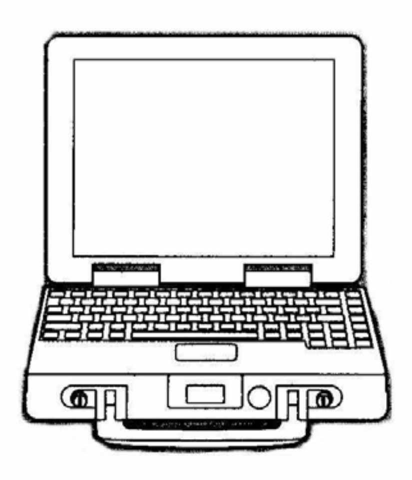
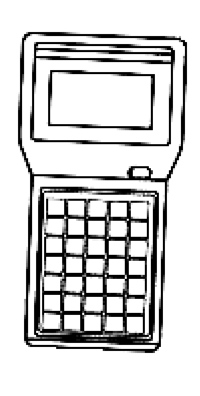


Fig. 19 PC portable





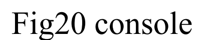


Fig20 console

Fig. 21 console

I.7 - Les étapes à suivre pour raccorder un automate:

Pour raccorder un automate, il est recommandé de suivre :

- Les spécifications du fabricant
- La technique de raccordement
- De vérifier si les modules sont dans leurs embases respectives. Vérifier le type, le numéro du modèle et le diagramme de câblage. Vérifier l'emplacement des embases dans le document pour l'assignation des adresses d'E/S.
- De localiser le paquet de fils correspondant à chaque module et le diriger à travers le conduit à l'emplacement du module.
  Identifier chacun des fils dans le paquet et s'assurer qu'ils correspondent à ce module en particulier.
- En commençant avec le premier module, repérer le fil dans le paquée qui se branche à la borne la plus basse. Au point où le fil arrive à la même hauteur que le point de terminaison, plie le fil à angle droit vers la borne.
- De couper le fil pour qu'il dépasse de 6 mm du côté de la vis de la borne. Dégainer l'isolant du fil à approximativement 9 mm
  Insérer le fil sous la plaque de la borne et serrer la vis.
- Si deux modules ou plus utilisent la même source d'alimentation, on peut utiliser du cavalier «jumpers » pour le câblage de la source d'alimentation d'un module à l'autre.
- Si le câble blindé est utiliser, en brancher seulement un bout à la mise à la terre, préférablement au châssis. Ce branchement évitera toutes boucles possibles de retour de masse. L'autre bout doit être coupé et non branché.
- De répéter la procédure de câblage pour chaque fil du paquet jusqu'à ce que le câblage du module soit complété. Après que tous les fils aient été branchés, tirer doucement sur chacun pour s'assurer d'avoir un bon branchement.

De répéter la procédure de câblage jusqu'à ce que tous les modules soient terminés.

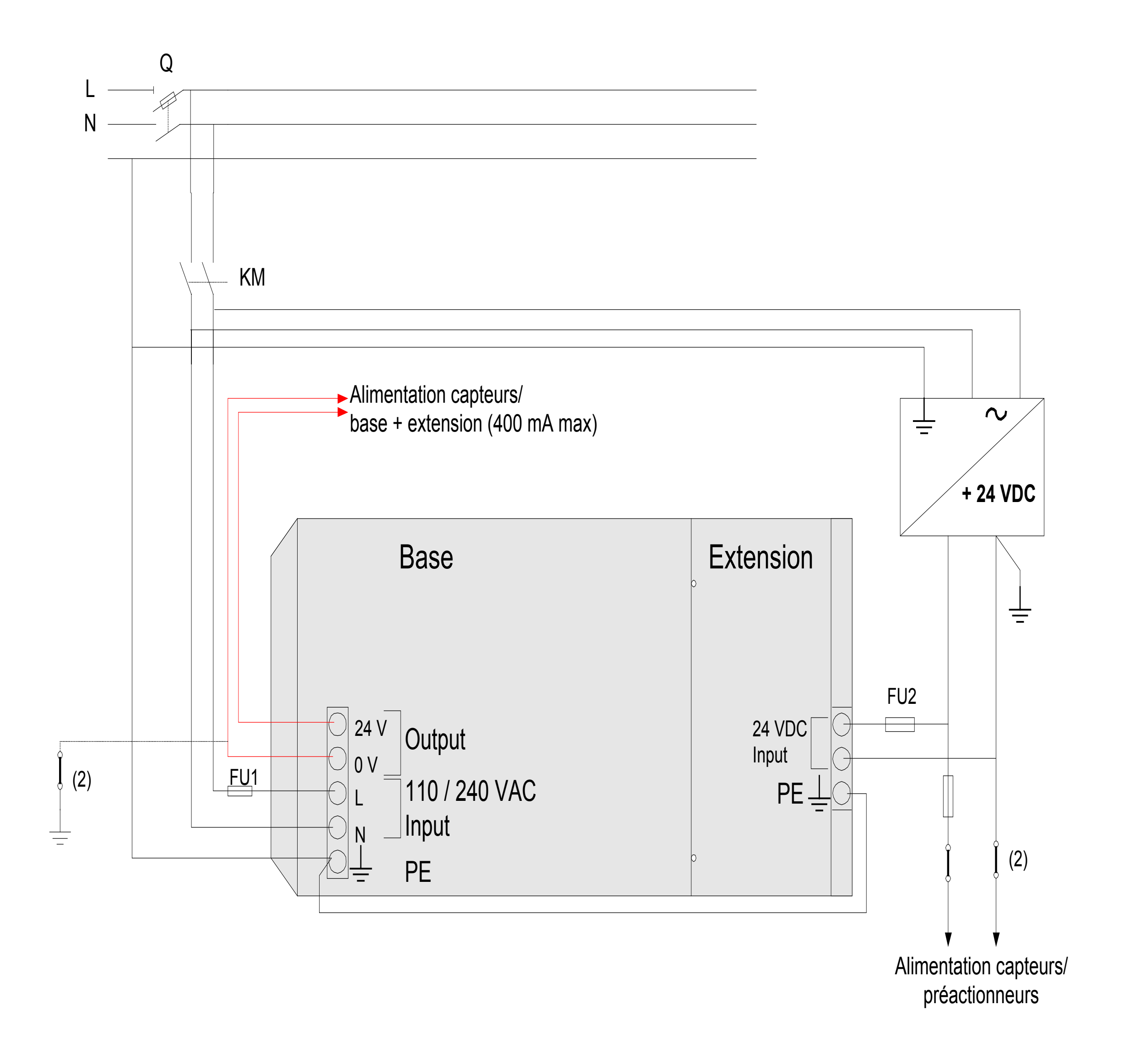


Fig 22 : Raccordement des alimentations :

II - ACCES AUX FONCTIONS D'UN AUTOMATE :

II.1 - Les Langages de programmation :

II.1.1 - Langage à contacts

Structure d'un programme :

Un programme en langage à contacts est composé d'une suite de réseaux de contacts exécutée de façon séquentielle par l'automate :

Dessiné entre deux barres de potentiel, un réseau est un ensemble d'éléments graphiques représentant :
• les entrées/sorties de l'automate (boutons-poussoirs, détecteurs, relais, voyants...),
• des fonctions d'automatismes (temporisateurs, compteurs...),
• des opérations arithmétiques, logiques et spécifiques,
• les variables internes de l'automate.
Ces éléments graphiques sont reliés entre eux par des connexions horizontales et verticales. (Voir figure 1)

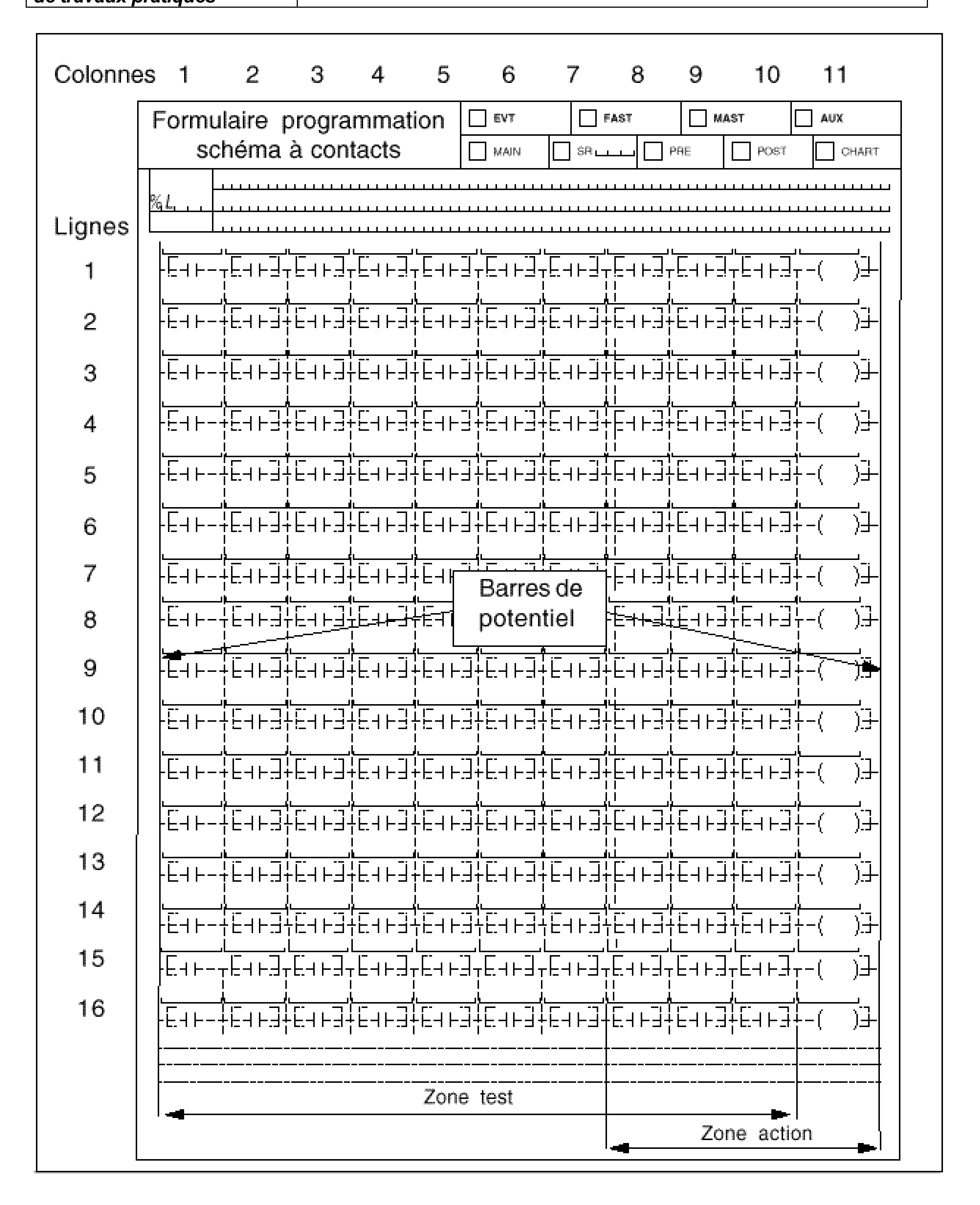


Fig 1 : Structure d'un réseau de contacts

EXEMPLE :

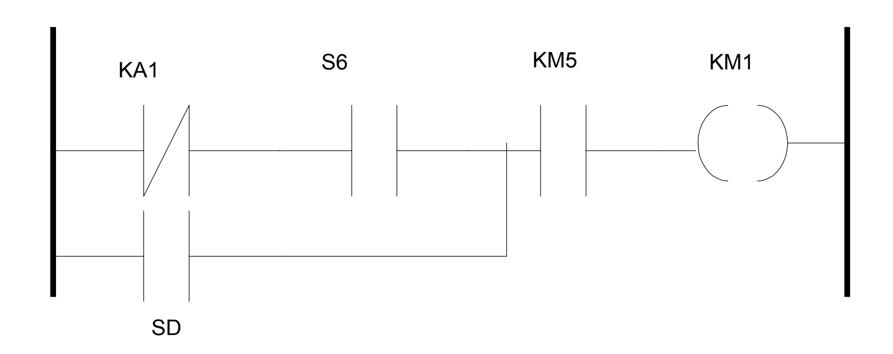


Fig 2 : Exemple écrit avec un langage à contacts

II.1.2 - Langage GRAFCET

Le **GRAFCET** est une représentation graphique qui permet la transcription du fonctionnement d'un système automatique. Il prend en compte les entrées et les sorties, et définit le comportement séquentiel du système.
L'**étape** correspond à une situation élémentaire ayant un comportement stable.
Une **transition** indique la possibilité d ' évolution d 'une étape à l' étape suivante. A chaque transition , on associe une, ou des conditions logiques qui traduisent la notion de **réceptivité**.

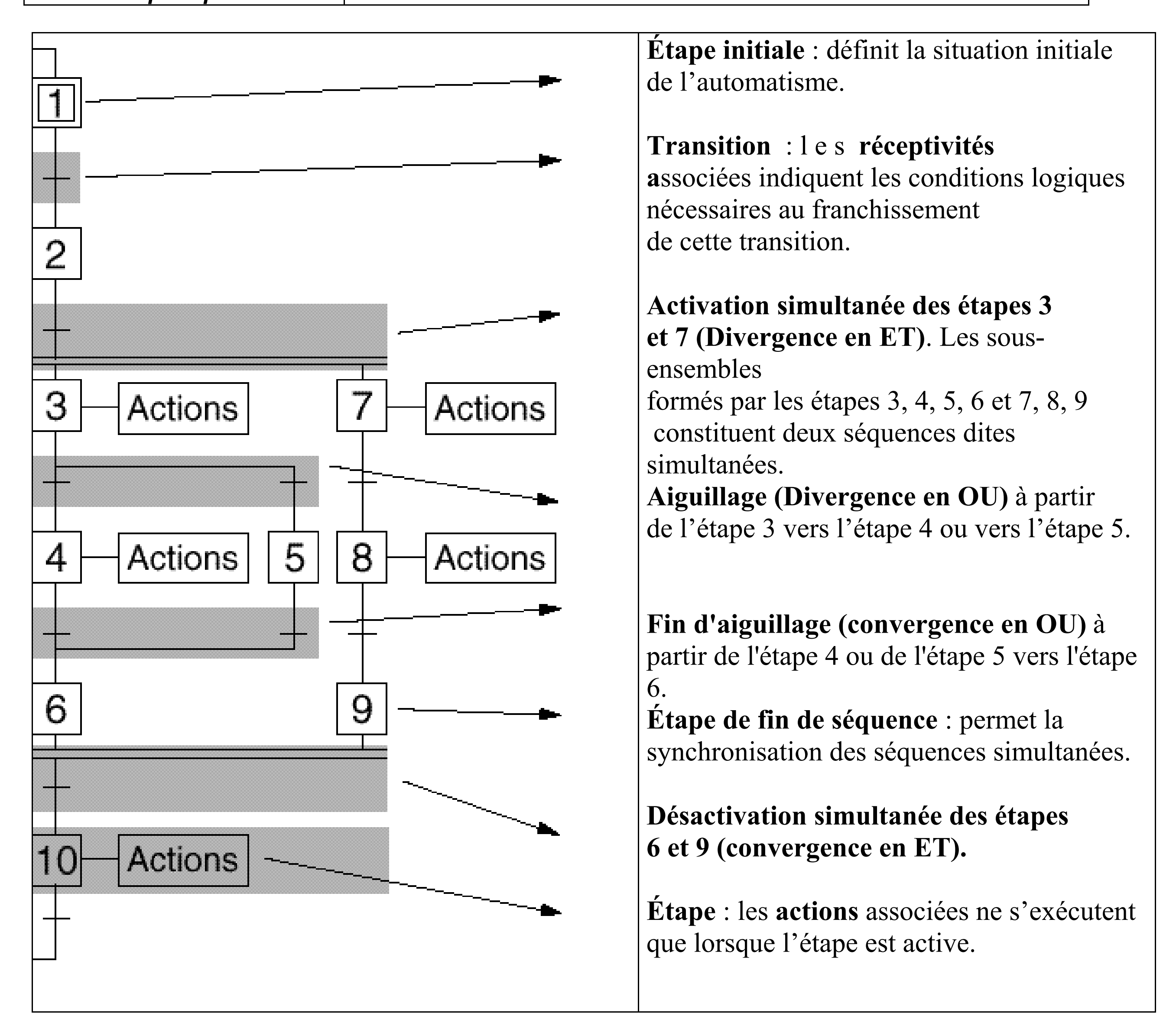
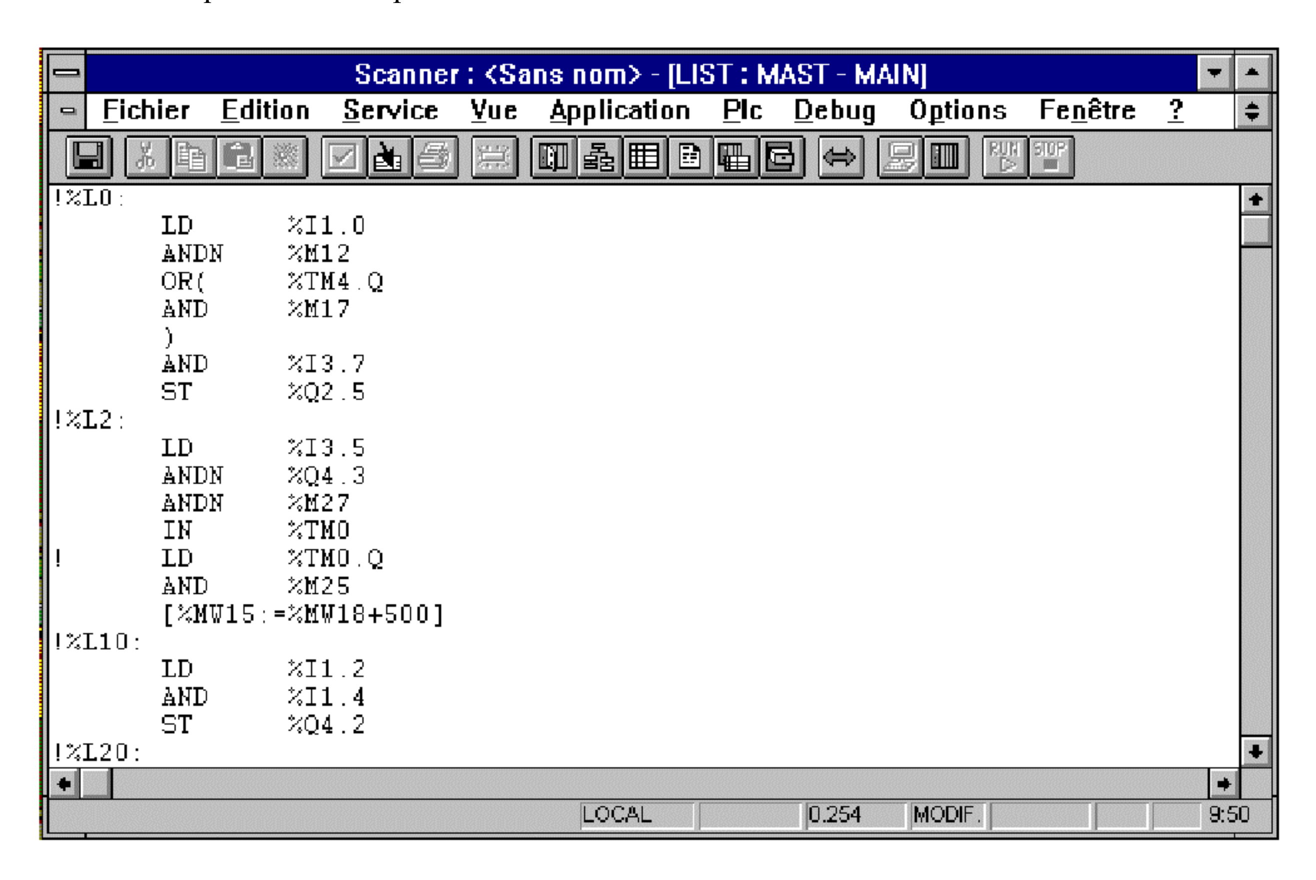


Fig 3 : exemple avec langage grafcet

II.1.3 - Présentation du langage liste d'instructions :

a) Principe

Un programme écrit en langage liste d'instructions se compose d'une suite d'instructions exécutées séquentiellement par l'automate.



Exemple d'instruction : LD %I1.0

Code instruction — Opérande

Chaque instruction est composée d'un code instruction et d'un opérande.
Ces instructions agissent sur :

- les entrées/sorties de l'automate (boutons-poussoirs, détecteurs, relais, voyants...),
- des fonctions d'automatismes (temporisateurs, compteurs...),
- des opérations arithmétiques et logiques et des opérations de transfert,
- les variables internes de l'automate.

Il existe 2 types d'instructions :

- Instruction de test, dans laquelle figurent les conditions nécessaires à une action, ex : LD, AND, OR...
- Instruction d'action, qui sanctionne le résultat consécutif à un enchaînement de test.
ex : ST, STN, R, ...

b) Les instructions
Instructions de base

Désignation	Instructions	Fonctions équivalentes
Instructions de test	• **LD, LDN, LDR, LDF**	
	• **AND, ANDN, ANDR, ANDF**	
	• **OR, ORN, ORR, ORF**	
	• **AND(, OR(** (8 niveaux de parenthèses)	
	• **XOR, XORN, XORR, XORF**	Ou exclusif
	• **MPS** **MRD** **MPP**	
	• **N**	Négation
Instructions d'action	• **ST, STN, S, R**	
	• **JMP, JMPC, JMPCN**	Permet un branchement (inconditionnel, conditionné à un résultat booléen à 1, conditionné à un résultat booléen à 0) à une instruction étiquetée, amont ou aval.
	• **SRn** **RET, RETC, RETCN**	Branchement en début de sous-programme Retour de sous-programme (inconditionnel, conditionné à un résultat booléen à 1, conditionné à un résultat booléen à 0).
	• **END, ENDC, ENDCN** **HALT**	Fin de programme, (inconditionnel, conditionné à un résultat booléen à 1, conditionné à un résultat booléen à 0). Arrêt de l'exécution du programme.

http://module01-ofppt.blogspot.com/

c) Structure d'un programme

Généralités

Comme en langage à contacts, les instructions sont organisées en séquence d'instructions (équivalent à un réseau de contacts) appelée phrase. Chaque phrase se compose d'une à plusieurs instructions de test, le résultat de ces instructions étant appliqué à une ou plusieurs instructions d'action.
Une instruction occupe une ligne maximum. Chaque phrase commence par un point d'exclamation (généré automatiquement), elle peut comporter un commentaire et être repérée par une étiquette.

```
! (*Attente de séchage*)
%L2:
LD      %I0.1
AND     %M10
ST      %Q2.5
```

Commentaire

Le commentaire peut être intégré au début d'une phrase et peut occuper 3 lignes maximum (soit 222 caractères alphanumériques), encadrés de part et d'autre par les caractères **(*** et ***)**. Il facilite l'interprétation de la phrase à laquelle elle est affectée, mais n'est pas obligatoire.
Les commentaires s'affichent uniquement à partir de la première ligne de la phrase. En cas de suppression d'une phrase, le commentaire qui lui est associé est également supprimé.
Les commentaires sont mémorisés dans l'automate et sont accessibles à tout moment par l'utilisateur. A ce titre, ils consomment de la mémoire programme

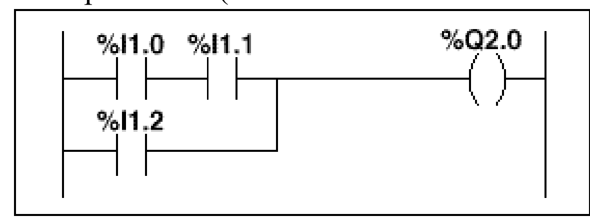
Étiquette

L'étiquette permet de repérer une phrase dans une entité de programme (programme principal, sous-programme, ...) mais n'est pas obligatoire.
Cette étiquette a la syntaxe suivante : %Li avec i compris entre 0 et 999 et se position en début d'une phrase.
Un repère d'étiquette ne peut être affecté qu'à une seule phrase au sein d'une même entité de programme.
Par contre il est nécessaire d'étiqueter une phrase afin de permettre un branchement après un saut de programme.
L'ordre des repères des étiquettes est quelconque, c'est l'ordre de saisie des phrases qui est prise en compte par le système lors de la scrutation.

Utilisation des parenthèses

Les instructions AND et OR peuvent utiliser des parenthèses. Ces parenthèses permettent de réaliser des schémas à contacts de façon simple. L'ouverture de parenthèses est associée à l'instruction AND ou OR. La fermeture de parenthèse est une instruction, elle est obligatoire pour chaque parenthèse ouverte.

Exemple : AND(




```
LD      %I1.0
AND     %I1.1
OR      %I1.2
ST      %Q2.0

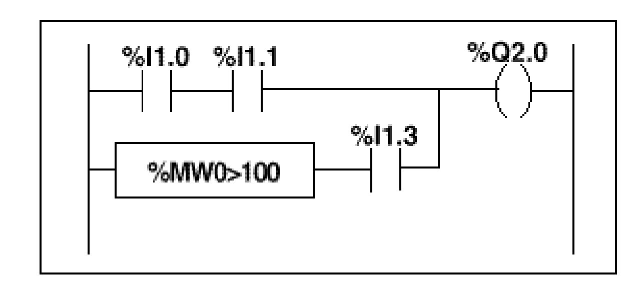
LD      %I1.0
AND(    %I1.1
OR      %I1.2
)
ST      %Q2.0
```

Exemple : OR(

```
LD      %I1.0
AND     %I1.1
OR(N    %I1.2
AND     %I1.3
)
ST      %Q2.0
```

Aux parenthèses peuvent être associées les modificateurs :

- N négation, ex : AND(N ou OR(N,
- F front descendant (**F**alling edge), ex : AND(F ou OR(F,
- R front montant (**R**ising edge), ex : AND(R ou OR(R,
- [comparaison.




```
LD      %I1.0
AND     %I1.1
OR(     [%MW0>100]
AND     %I1.3
)
ST      %Q2.0
```

Imbrication de parenthèses

Il est possible d'imbriquer jusqu'à 8 niveaux de parenthèses.

Exemple

```
LD      %I1.0
AND(    %I1.1
OR(N    %I1.2
AND     %M3
)
)
ST      %Q2.0
```

Exemple

```
LD      %I1.1
AND(    %I1.2
AND     %I1.3
OR(N    %I1.5
AND     %I1.6
)
AND     %I1.4
OR(N    %I1.7
AND     %I1.8
)
)
ST      %Q2.0
```

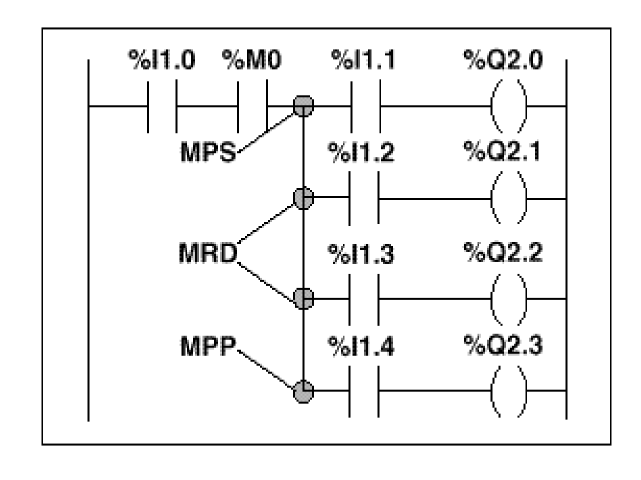
Note :

- chaque parenthèse ouverte doit être impérativement refermée.
- les étiquettes %Li: ne doivent pas être placées dans des expressions entre parenthèses, ainsi que les instructions de saut JMP et d'appel à sous programme SRi,
- les instructions d'affectation ST, STN, S et R ne doivent pas être programmées entre parenthèses.

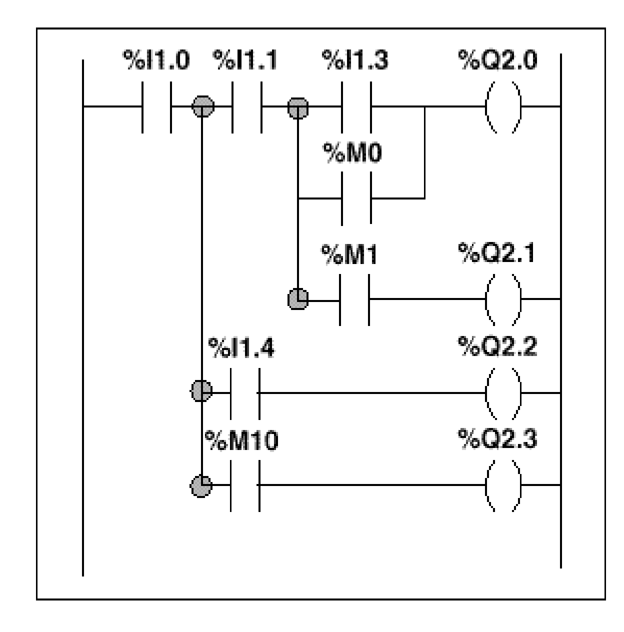
Instructions MPS, MRD, MPP

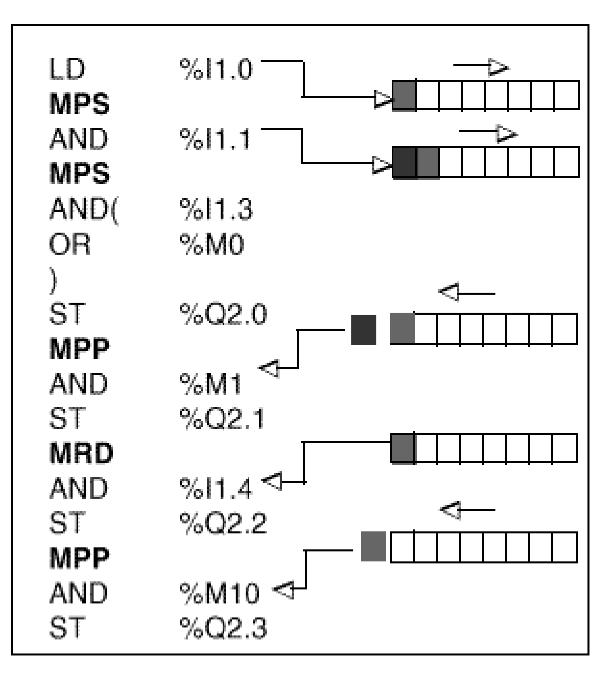
Les 3 types d'instruction permettent de traiter les aiguillages vers les bobines.
Ces instructions utilisent une mémoire intermédiaire appelée pile pouvant stocker jusqu'à 3 informations booléennes.
L'instruction MPS (Memory PuSh) a pour effet de stocker le résultat de la dernière instruction de test au sommet de la pile et de décaler les autres valeurs vers le fond de la pile.
L'instruction MRD (Memory ReaD) lit le sommet de la pile.
L'instruction MPP (Memory PoP) a pour effet de lire, de déstocker le sommet de la pile et de décaler les autres valeurs vers le sommet de la pile.

Exemples :




```
LD      %I1.0
AND     %M0
MPS
AND     %I1.1
ST      %Q2.0
MRD
AND     %I1.2
ST      %Q2.1
MRD
AND     %I1.3
ST      %Q2.2
MPP
AND     %I1.4
ST      %Q2.3
```





http://module01-ofppt.blogspot.com/

Programmer une instruction

Les automates TSX sont programmables en langage LIST : listes d'instructions ou LADDER : langage à contacts.

Représentation en langage à contacts	Représentation en liste d'instructions	Interprétation
.1	**LD %I1.1**	Lire l'état de l'entrée %I1.1 **« Examine si c'est fermé »**
.1	**LDN %I1.1**	Lire l'état inverse de l'entrée %I1.1 **« Examine si «c'est ouvert. »**
2.1	**ST %Q2.1**	Transférer le résultat dans la sortie 1
2.1	**STN %Q2.1**	Transférer l'inverse du résultat dans la sortie 1

Instruction « AND » (« ET ») : transcription de deux ou plusieurs contacts en série

Représentation en langage à contacts	Représentation en liste d'instructions	Interprétation
.1	**LD %I1.1** **AND %I1.2**	Lire l'état des entrées 1 ET 2

Instruction « ANDN » (« ET NON ») :

Représentation en langage à contacts	Représentation en liste d'instructions	Interprétation
.1 .2	**LD %I1.1** **ANDN %I1.2**	Lire l'état des entrées 1 ET l'état inverse de l'entrée 2

Instruction « OR » («OU») :

Représentation en langage à contacts	Représentation en liste d'instructions	Interprétation
.1 .2	LD %I1.1 **OR %I1.2**	Le résultat de cette combinaison est égal à 1 si l'entrée 1 OU l'entrée 2 est à 1 OU si les 2 entrées sont à 1

Instruction « ORN » («OU NON») :

Représentation en langage à contacts	Représentation en liste d'instructions	Interprétation
.1 .2	**LD %I1.1** **ORN %I1.2**	Le résultat de cette combinaison est égal à 1 si l'entrée 1 OU l'entrée 2 est à 0

II.2 - Les principales instructions d'un automate :

Les principales instructions avec lesquelles ont peut programmer aisément se résument dans ce tableau :

Désignation	Symbole	Fonction
Les contacts	—| |— —|/|— —|P|— —|N|—	Contact ouvert au repos Contact fermé au repos Contact active au front montant Contact active au front descendant
Les bobines	—( )— —(/)— —(S)— —(R)—	Bobine directe Bobine inverse Bobine d'enclenchement Bobine de déclenchement
Blocs fonctions	- Temporisateur %TMi IN Q T_{ON} IN Q T_{OF} - Compteur / decompteur %Ci R E S D Cu F CD E : Bit sortie débordement D : Bit sortie présélection atteinte F : Bit sortie débordement (9999->0)	T_{ON} : retard à l'enclenchement T_{OF} : retard déclenchement IN : Entrée Q : Sortie temporisateur % TM : label du temporisateur. % Ci : label du compteur / decompteur R : Entrée remise à zéro. S : Entrée de présélection. Cu : Entrée incrémentation sur front. CD : Entrée décrémentation sur front

	- Séquenceur ou programmateur cyclique. %Dri R F U LEN	R : Remise à zéro U : Entrée avance pas. LEN : Nombre de pas.
Blocs opérations	%MW0 = %MW10 + 100	Ce bloc permet d'effectuer toutes les autres opérations à savoir : - Addition - Sous traction - Division - Multiplication
Blocs comparaison	% MW20 ≥ 25	Ce bloc permet d'effectuer toutes les comparaisons

II.3 - L'utilisation d'un logiciel de programmation :

Pour pouvoir utiliser le logiciel :

1- Il faut d'abord commencer par l'installation du logiciel de programmation ,pour cela on met le CD ROM dans le lecteur de CD ROM et suivre les démarches habituelles d'installation soit sous Windows ou sous dos (Exécuter, Parcourir, etc.…)

2- Configurer le matériel c'est à dire spécifier l'automate avec lequel on va travailler en donnant sa référence .

3- On choisit le mode de programmation en choisissant l'éditeur approprie : Ladder (à contact) ; (LD) langage structurée (LS), ou Grafcet, (CHART).

4- Le mode en ligne consiste à passer l'application pour que l'automate l'exécute. Pour cela le logiciel offre cette possibilité qui permet de passer du Mode local en mode connecté en appuyant sur l'icône Connecter, puis Transférer.

II.4 – Les moyens d'accès aux fonctions d'un automate :

Les moyens d'accès aux fonctions d'un automate consiste à configurer l'automate, voir comment est structuré les mémoire et enfin comment connaître les paramètres de communication.

- La procédure qu'on énoncer concerne le logiciel PL7 Micro adapté à l'automate TSx3708 et version 3.0
- On commence par ouvrir une nouvelle application.
- Dans le menu Fichier on appuie sur nouveau.
- On passe à la configuration matérielle en choisissant l'automate dans une série offerte par logiciel.
- Pour allouer l'espace nécessaire à l'application.
- On pourra voir dans le menu AP ; le menu Bilan mémoire qui nous montre comment la mémoire est partagé.

Configuration des E/S

Le logiciel doit permettre de sélectionner la position et le format du module(format standard, demi-format) à configurer soit par clic sur la souris ou en manipulant les touches flèches du clavier. La configuration logicielle permet de définir pour l'application :

--Le nombre des différents types de blocs de fonctions
--Le nombre de mots registres,
--Le nombre de bits internes °/°Mi,
--Le nombre de mots internes °/° MW,
--Le nombre de constantes °/° KW.

1/ Ecriture des adresses des E/S :

Bit = % I 1.O %Q 2.3
Mot = % IW 6.2 °/° QW6.O

2 / Ecriture des adresses des Bit internes

$\%B_1$

3 / Écriture des adresses des Le bloc fonction

°/° T1
°/° C1

II.5 - La méthode de programmation : (Résumé)

La méthode proposée vous permet de programmer, tester, mettre au point et sauvegarder votre programme.

Étape 1 : **Configuration de l'application**

Nom de l'application, valeurs des constantes, horodateur, paramètre des compteurs, temporisateurs.

Étape 2 : **Saisie des symboles**

Saisie des noms de Symboles pour chaque repère utilisé dans votre programme automate (contacts, bobines…).

Étape 3 : **Saisie du programme**

Et validation des saisies.

Étape 4 : **Sauvegarde régulière du programme**

En cours de saisie –PC vers disque.

Étape 5 : **Transfert du programme dans l'automate**

Étape 6 : **Mise en RUN et teste du programme**

Étape 7: **Mise au point du programme**

Éditeur de données.

Étape 8 : **Sauvegarde du programme**

Après mise au point – Automate ➞ disque

Étape 9 : **création du dossier de l'application**

III - Diagnostic des problèmes de fonctionnement d'un automatisme simple commandé par un automate :

III.1 - Visualisation centralisée :

Les automates sont équipés d'un bloc de visualisation centralisant toutes les informations nécessaires au contrôle, au diagnostic et à la maintenance de l'automate et de ses modules, et des fonctions simples de dialogue opérateur. Les automates sont équipés des blocs de visualisation qui se différent selon la marque(l' éxemple.pris est celui de marque Télémécanique.)

La visualisation centralisée offre :

- La visualisation de l'état des voies d'entrées / sorties locales ou distantes (entrées / sorties des automates Nano).
- La visualisation des équipements sur le bus AS-i et le diagnostic de ce dernier (voir page 2/60)
- Le diagnostic des voies ou des modules en défaut.
- La visualisation de données internes :
  - Bits,
  - Chaînes de bits,
  - Chaînes de mots,
  - Variables du programme (étapes actives, informations d'application…)
- Une visualisation numérique multiple sur 4 digits.

Description

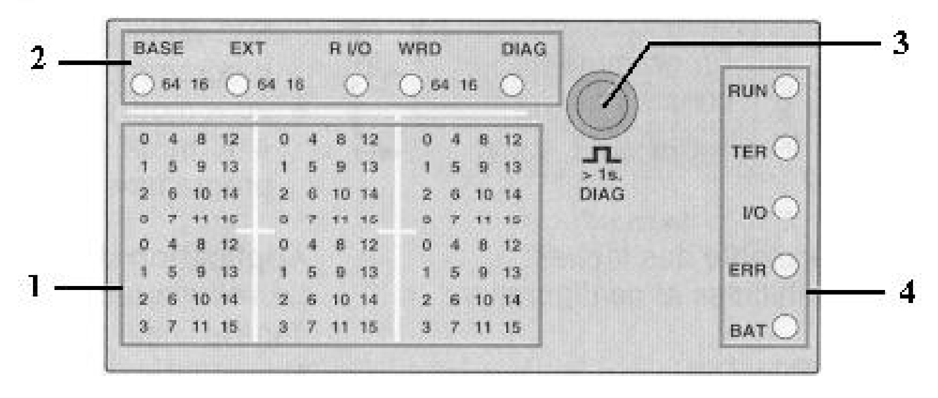


Fig. 1 Bloc de visualisation centralisée

Le bloc de visualisation centralisée comprend :

1. Trois ensembles de 32 voyants (DEL) représentant les emplacements des modules implantés dans le bac de base ou le mini bac d'extension.
2. Une ligne d'information formée de voyants (DEL) signalant les modes de fonctionnement de la visualisation.
3. Un bouton-poussoir de commande donnant accès aux différents modes de fonctionnement de la visualisation.

4. Cinq voyants (DEL) :
   - RUN, marche / arrêt de l'automate,
   - TER, trafic sur la prise terminal,
   - I/O, défaut processeur ou application,
   - BAT, défaut ou absence de pile
   -

III.2 - Les problèmes de fonctionnement d'un automate programmable :

- Méthode générale de dépannage d'un automate simple commandé par un automate.

- Tout système est divisé en quatre blocs :
  - Bloc 1 : alimentation ;
  - Bloc 2 : commande ;
  - Bloc 3 : puissance ;
  - Bloc 4 : sortie.

- Faire une vérification visuelle des composants :

- Si le composant défectueux est facilement repérable, le remplacer. Faire une vérification obligatoire des causes du défaut avant de passer à l'étape suivant.
- Si le composant défectueux est difficilement repérable, aller directement à l'étape suivante.
- S'il n'y a pas de danger, débranchement de la sortie du système et branchement sur une charge factice («dummy load ») avant de procéder à un essai de mise en marche.

- Vérification de l'état des composants durant essai (vue, odorant, ouïe, toucher).

- Si rien ne fonctionne, vérification des blocs selon l'endroit où le défaut est le plus susceptible de s'être produit, 1, 4, 3, 2.

- En cas de fonctionnement partiel, vérification des blocs selon l'endroit où le défaut est le plus susceptible de s'être produit, 4, 3, 1, 2.

- Une fois la réparation effectuée, essai de fonctionnement du système pendant un temps suffisant pour permettre de conclure que le système est fonctionnel.

- Application des normes du fabricant (attendre le temps nécessaire pour que tous les appareils entrent en action). Poursuite de l'attente jusqu'à ce que les appareils atteignent leur température de fonctionnement de manière à s'assurer que le système ne tombe pas en panne à cause d'une dérive thermique des composants.

- S'il faut débrancher un composant, s'assurer que toutes les alimentations sont hors fonction.
- Les appareils de mesure suggérés pour le dépannage du système automatisé sont : ordinateur, voltmètre, pince ampèremétrique et oscilloscope (éviter le plus possible l'utilisation de l'ampèremètre série pour minimiser les risques de défaut par mauvais branchement). L'ohmmètre devrait être utiliser avec beaucoup de précautions.

- Utiliser le caractéristique de l'automate en mode :
- Manuel ;
- Automatique ;
- Étape par étape.

Décoder les messages d'erreurs

III.3 - Les modifications apportées au programme d'un automate :

L'avantage de la logique programmé par rapport à la logique câblée c'est qu'elle offre la possibilité de modifier la programme telle que :

- La modification de la valeur de présélection d'un temporisateur, d'un compteur / dé compteur.
- Changer un contact fermé ou ouvert par un autre type de contact.
- Ajouter une bobine, ou un sous programme etc.

IV - L'essai d'un automatisme simple commandé par un automate :

IV.1 - Les dangers potentiels :liés à l'utilisation d'un automate :

Le forçage des entrées / sorties consiste à mettre à 1 le bit image de ces dernières grâce au logiciel de programmation.

Ou procède au forçage lorsque :

- Ou ne dispose pas matériellement des entrées ou sorties. (absence d'un bouton poussoir pour la mise en marche d'un engin, absence d'un contacteur…)
- On veut déceler les défauts provenant E/S.
- On veut voir l'évolution de la programmation du processus automatisme avant de passer au câblage.

Le forçage est prioritaire, il est conseillé de l'utiliser avec précaution car il présente certains dangers.

Exemple 1 :

Lorsqu'on force à 1 un sortie automate qui commande l'ouverture d'une vanne évacuant de l'air chaude, celle-ci peut brûler les travailleurs, endommager le matériel autour.

Exemple 2 :

Le forçage d'une entrée commandant la fermeture, automatisée de l'issue de secours, ou la sortie d'un vérin.

Ajoutons que le verrouillage par logiciel s'avère insuffisant comme moyens de sécurité.

Il est indispensable de procéder à un verrouillage matériel ainsi qu'a la procédure de cadenassage parfois car il est plus sécuritaire.

IV.2 - L'essai d'un automatisme simple:

On doit confirmer qu'un automatisme commandé par automate est fonctionnel, après avoir lancé l'exécution et vérifier que :

- L'ensemble fonctionne pendant une durée assez suffisante permettant d'atteindre les températures de fonctionnement afin de s'assurer que le système est fonctionnel et qu'il ne va pas tomber en panne à cause d'une dérive thermique de composants.

- Si on effectue un essai après avoir changer un ou plusieurs composants défectueux, on vérifie que tout entre en action après le changement effectué.

Module24 : Utilisation De l'Automate Programmable
GUIDE DES EXERCICES ET TRAVAUX PRATIQUES

I - Raccordement d'un automate programmable.

EXERCICE 1 :

Question n°1 :

Décrire cet automate en nommant ces différentes parties ?

- N°1=
- N°2=
- N°3=

N°4=

- N°5=
- N°6=
- N°7=
- N°8=

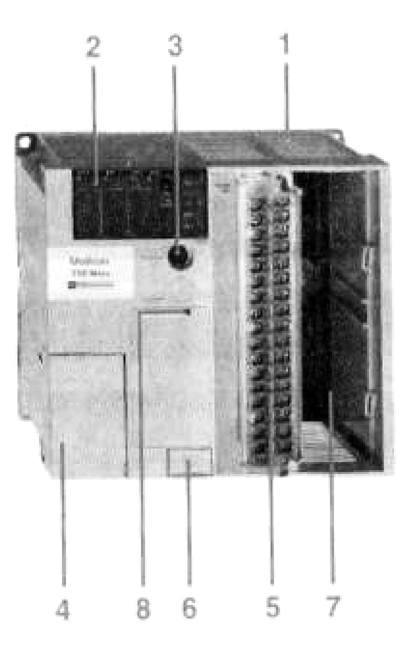


Fig 1

Question n°2 :

Nommer les différentes parties de cet automate monobloc?

- N°1=
- N°2=
- N°3=
- N°4=
- N°5=
- N°6=
- N°7=
- N°8=

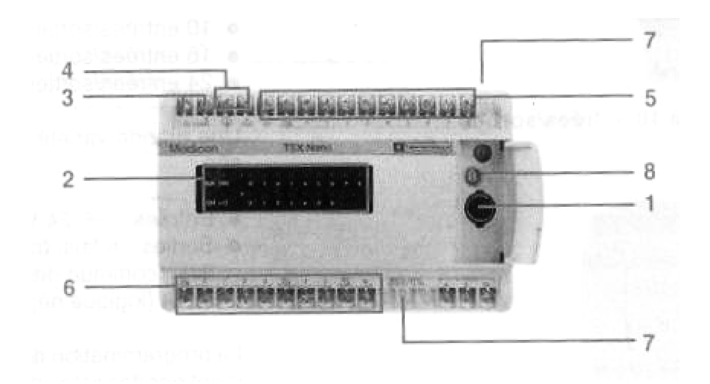


Fig. 2

EXERCICE 2 :

Question : Reconnaître parmi ces figures les modules analogiques, des modules tout ou rien?(entrée/sortie)

1=entrée……………………..? **Fig.1**
Module =……………………? **Fig.2**

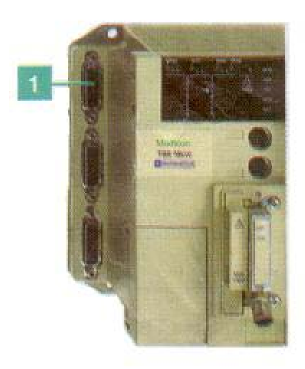


Fig. 1





Fig. 2

EXERCICE 3 :

Nommer le logiciel approprié à chaque marque d'automate suivant ?:

Marque	Automate	Logiciel
Télémécanique	TSXNano	?
Siemens	?	Step7

I. TP 1 : raccordement d'un automate :

I.1. Objectif(s) visé(s) :

- Distinguer les entrées des sorties de l'automate
- Câbler l'automate

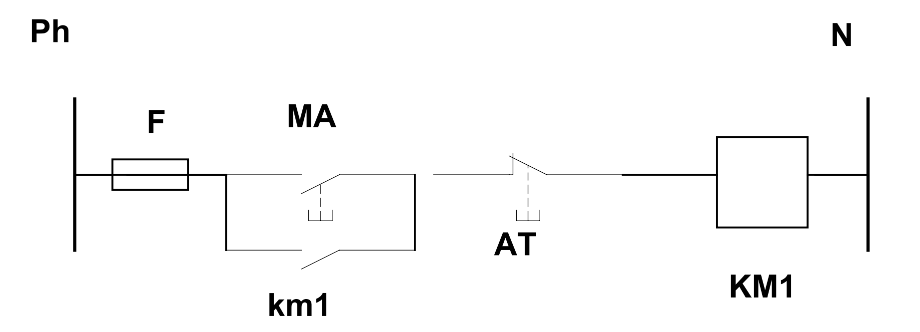
I.3. Matériel (Équipement et matière d'œuvre) par équipe :

Équipement *:*

- Un automate
- Un contacteur
- Un bouton poussoir marche et un bouton poussoir arrêt

I.4. Description du TP :

- Câbler sur l'automate l'exemple suivant d'un démarrage direct :



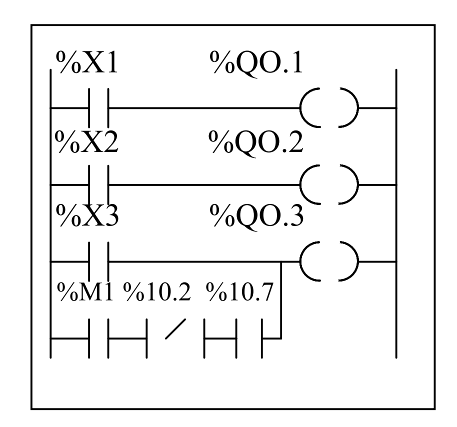
- Distinguer les entrées des sorties et câbler en remplissant le tableau.

Désignation	Fonction	Adresse sur automate
BP BA KM		

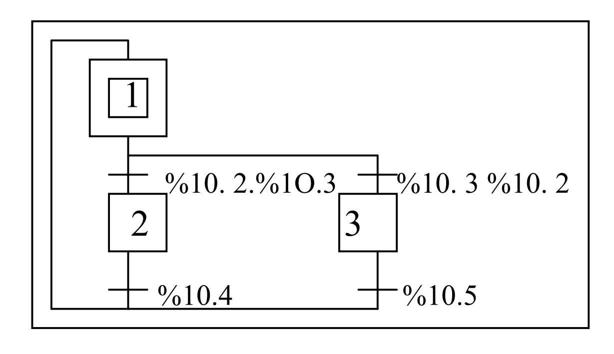
II - Accès aux fonctions d'un automate.

EXERCIC 1 :

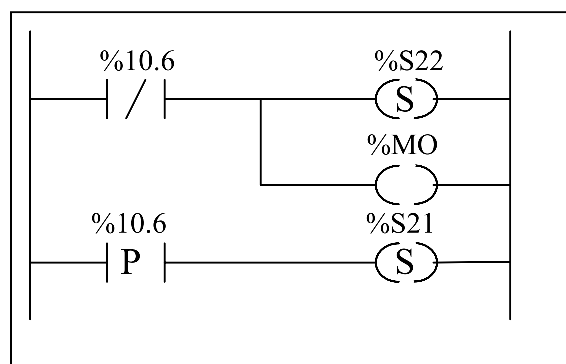
Question 1 : Distinguer entre les langages de programmation.



Langage…………

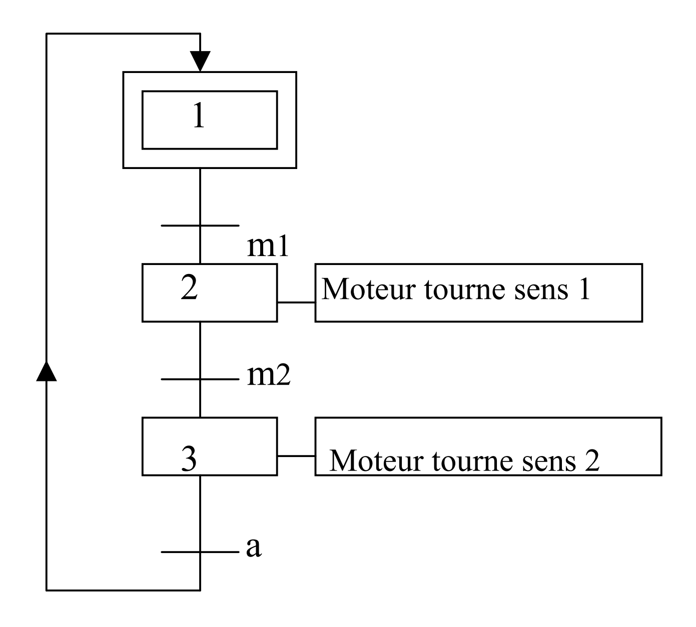


Langage……………



Langage…………………….

Question 2 : On donne l'exemple suivant :



Que représente :

Le graphe : Réponse :

EXERCIC 2 :

Question : Remplir le tableau suivant :

Symbole	Fonction
P	
%TMi IN Q	
%Ci R S Cu CD E D F	
(/)	

Que représente :

─┤ ├─	Réponse :
─┤ / ├─	Réponse :
─( )─	Réponse :

EXERCIC 3 :

Question n°1 :

Donner la marque et la référence de l'automate utilisé ?

Question n°2 :

Comment allez-vous procéder pour écrire le syntaxe d' :
- Une entrée automate ;
- Une sortie automate ;
- Un bit système ;
- Un bit interne ;
- Un bloc fonction temporisateur ?
-

Question 3 :

Que signifie les instructions suivantes :

-| |-	
-|/|-	
-( )-	
-(/)-	
-( )-	
-(R)-	
-| |-	

Question 4:

Cocher la bonne réponse :

Désignation	Entrée automate	Bit mémoire	Sortie automate	Mémoire vive	Mémoire morte
Lampe rouge					
Bouton poussoir arrêt					
ROM					
RAM					
Contact de relais thermique(95-96)					
PCMCIA					

Question 5:

Donner la définition des objets suivants :

%I1.0 :……………………………………………………

%M12 :……………………………………………………

%Q204.61 :……………………………………………

%TM1.Q :………………………………………………

%TM2.V :………………………………………………

SR0 :……………………………………………………

%Q4.5 :4 ………………………………………………….

EXERCICE 4:

Faites une programmation en ladder de charge de charge suivant.
Un chariot initialement à gauche effectue le déplacement suivant :

- déplacement à droite jusqu'à fin course droite
- retour à sa position initiale

Entrées :

- Fin course droite
- Fin course gauche
- Bouton poussoir marche
- Bouton arrêt

Sorties :

- Bobine gauche
- Bobine droite

TP: Utilisation d'un logiciel de programmation

I.1. Objectif(s) visé(s) :

- Configuration de l'automate
- Adressage des E/ S
- Le choix du mode de programmation
- Le passage en mode en ligne

I.2. Matériel (Équipement et matière d'œuvre) par équipe :

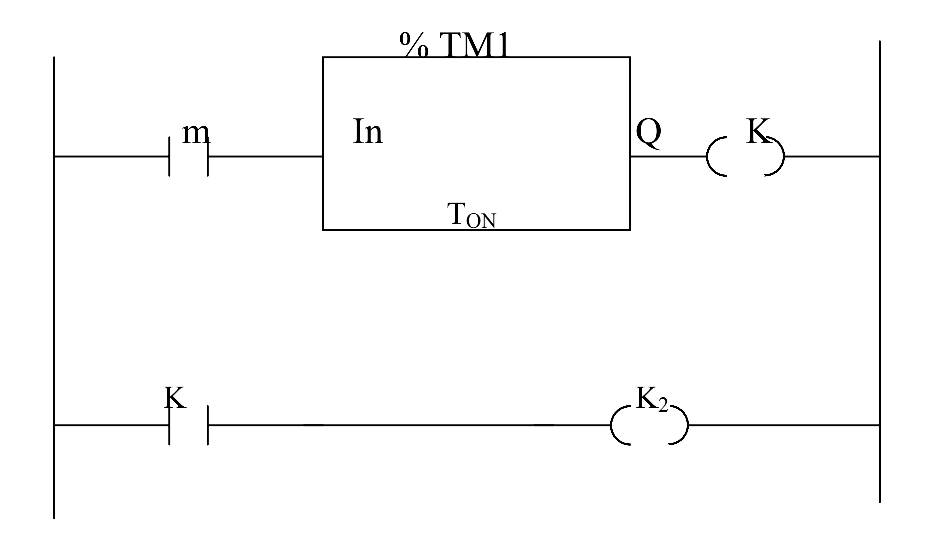
Équipement *:*

- Un automate
- UN P C compatible

I.3. Description du TP1 :

<u>Question N°1</u> : Décrire la démarche d'installation de logiciel de programmation utilisé ?

- **<u>Question N°2</u> :** Programmer le réseau suivant



Fixer le temps de présélection du temporisateur à 50 ms.

<u>Question</u> 3: Comment vous passer en mode RUN (exécution) ?

I.4. Description du TP2 :

A)

1/ **L**ancez le PL7 MICRO .

2/ **C**réez une nouvelle application automate en lui donnant un nom .

3/ **C**hoisissez un processeur TSX37XX (Activez l'option Grafcet) .

4/ **F**aites une configuration matériel en tenant compte des modules d'entrée - sortie existantes et de leur emplacement.

5/ **F**aites une configuration logicielle pour définir le nombre de temporisateurs , monostables ,

compteurs aussi la taille des zones de variables .

6/ **F**aites une configuration des objets Grafcet .

B)

1/ **F**aites activer l'option démarrage automatique en **RUN** de l'automate en mode non connecté/configuration.

2/ **F**aites régler l'horodateur de l'automate en mode connecté/mise au point .

3/ **E**ssayez d'enlever la pile de l'automate (opération à faire hors tension) ,consultez le **DIAG**
du mise au point (sous tension et en mode connecté/mise au point) qu'est ce qu'on on remarque ?.

I.5. Description du TP3 :

1/ Créer une nouvelle section en langage ladder (LD) en lui donnant un nom .

2/ **I**nstructions de chargement booléennes:

2-1/ **V**isualiser l'état de l'entrée %I3.0 sur la lampe de sortie %Q3.8

2-2/ **V**isualiser l'état inverse de l'entrée %I3.1 sur la lampe de sortie %Q3.9

2-3/ **V**isualiser le passage à l'état 1 de l'entrée %I3.2 sur la lampe de sortie %Q3.10

2-4/ **V**isualiser le passage à l'état 0 de l'entrée %I3.3 sur la lampe de sortie %Q3.11

I.6. Description du TP4 :

Durée du TP4:

1/ Instructions d'affectation booléennes :

1-1/ **A**ffecter l'état de %I3.4 à la sortie %Q3.12

1-2/ **A**ffecter l'inverse de l'état de %I3.5 à %Q3.13 et son état à %Q3.12

1-3/ **M**émoriser le passage à l'état 1 de %I3.6 dans %Q3.14

1-4/ **M**émoriser le passage à l'état 0 de %I3.7 dans %Q3.15

I.7. Description du TP5 :

1/

1-1/ **R**éaliser un programme en langage ladder pour une commande de marche manuelle d'un Moteur à un seul sens de marche M1 .

Arrêt = %I3.0
Marche = %I3.1
Bobine = %Q3.8

1-2/ Prendre en compte la signalisation de l'état du moteur :

- Sur une lampe l'état de marche = %Q3.9

- Sur une autre lampe l'état d'arrêt = %Q3.10

2/

2-1/ **R**éaliser un programme en langage ladder pour une commande de marche manuelle d'un Moteur à deux sens de marches (avant et arrière) M2 .

Arrêt = %I3.2
Marche avant = %I3.3
Marche arrière = %I3.4

Commande marche_av = %Q3.11

Commande marche_ar = %Q3.12

2-2/ **P**rendre en compte la signalisation de l'état du moteur :

- Sur une lampe verte l'état de marche _ avant = %Q3.13

- Sur une lampe orange l'état de marche _ arrière = %Q3.14

- Sur une lampe rouge l'état d'arrêt = %Q3.15

I.8. Description du TP6 :

Nous avons une grande machine qui a une porte de sécurité qui fait accès à un moteur central dans cette machine M

1/ **L**a commande du moteur se fait par un bouton de marche et un bouton d'arrêt .
L'ouverture de la porte de sécurité doit bloquer le démarrage du moteur M

2/ **S**i au cours du fonctionnement , une personne ouvre la porte de sécurité :

2-1/ Le moteur M doit s'arrêter .

2-2/ Une alarme doit se déclencher.

2-3/ Cette alarme doit être acquitter par l'opérateur par un bouton pousoir.

3/ **N**ous voulons que l'opérateur puisse arrêter le moteur en cas d'urgence par des arrêts d'urgences situés aux différents points de l'usine (quatre arrêts d'urgences) et que le moteur ne démarre que si on relâche l'arrêt d'urgence.

4/ **N**ous désirons que l'opérateur puisse localiser lequel des arrêts d'urgence est activé pour pouvoir l'éliminer pendant les conditions normaux .

BP marche = %I3.0
BP arrêt = %I3.1
FDC Ouverture Porte = %I3.2

BP acquittement défaut = %I3.3

Arrêt d'urgence n° 1 = %I3.4

Arrêt d'urgence n°2 = %I3.5

Arrêt d'urgence n°3 = %I3.6

Arrêt d'urgence n°4 = %I3.7

Commande moteur M = %Q3.8

Alarme = %Q3.9

Signalisation A.U 1 =%Q3.11

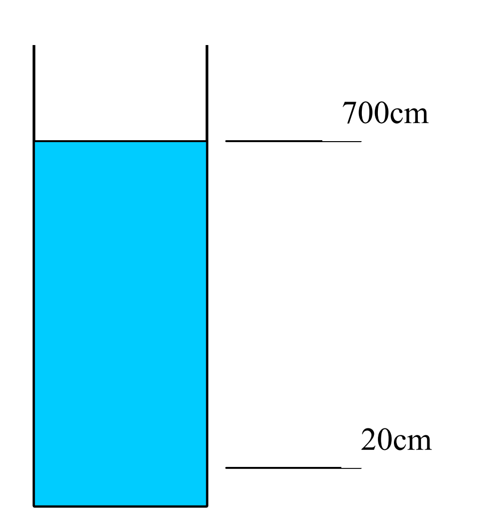
Signalisation A.U 2 =%Q3.12

Signalisation A.U 3 =%Q3.13

Signalisation A.U 4 =%Q3.14

I.9. Description du TP7 :

On mesure le niveau de liquide dans une cuve à l'aide d'une **entrée analogique** configurée en 0-10V.



Le niveau haut (700 cm) correspond à une valeur lue =10000 (10V) sur l'entrée analogique .

Le niveau bas (20cm) correspond à une valeur lue = 0 sur l'entrée analogique.

Il vous est demandé de calculer le niveau de la cuve en cm en fonction de la valeur lue (0-10000 mV) sur l'entrée analogique .

Essayez d'utiliser un potentiomètre pour simuler le signal 0-10V du capteur de niveau , est un voltmètre à la sortie pour visualiser la sortie analogique , suivant le câblage suivant :

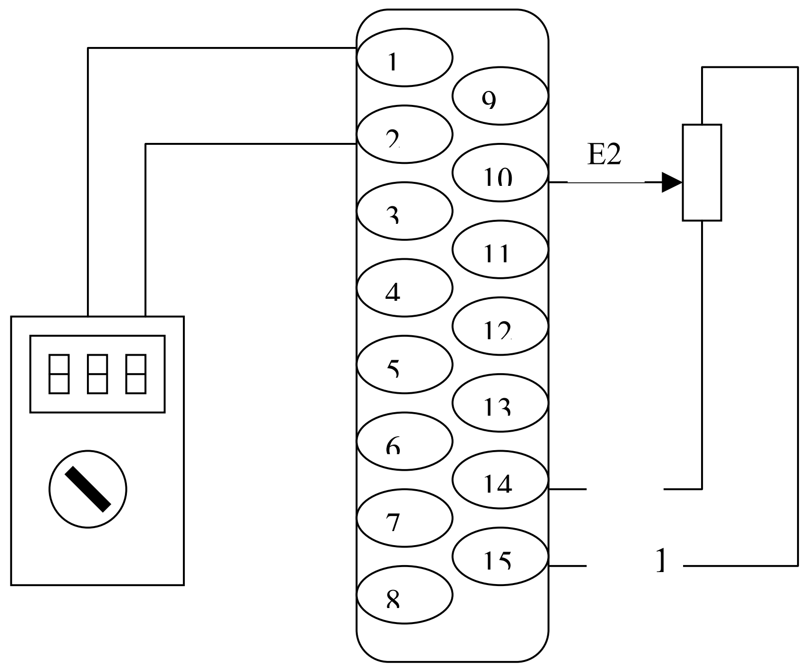
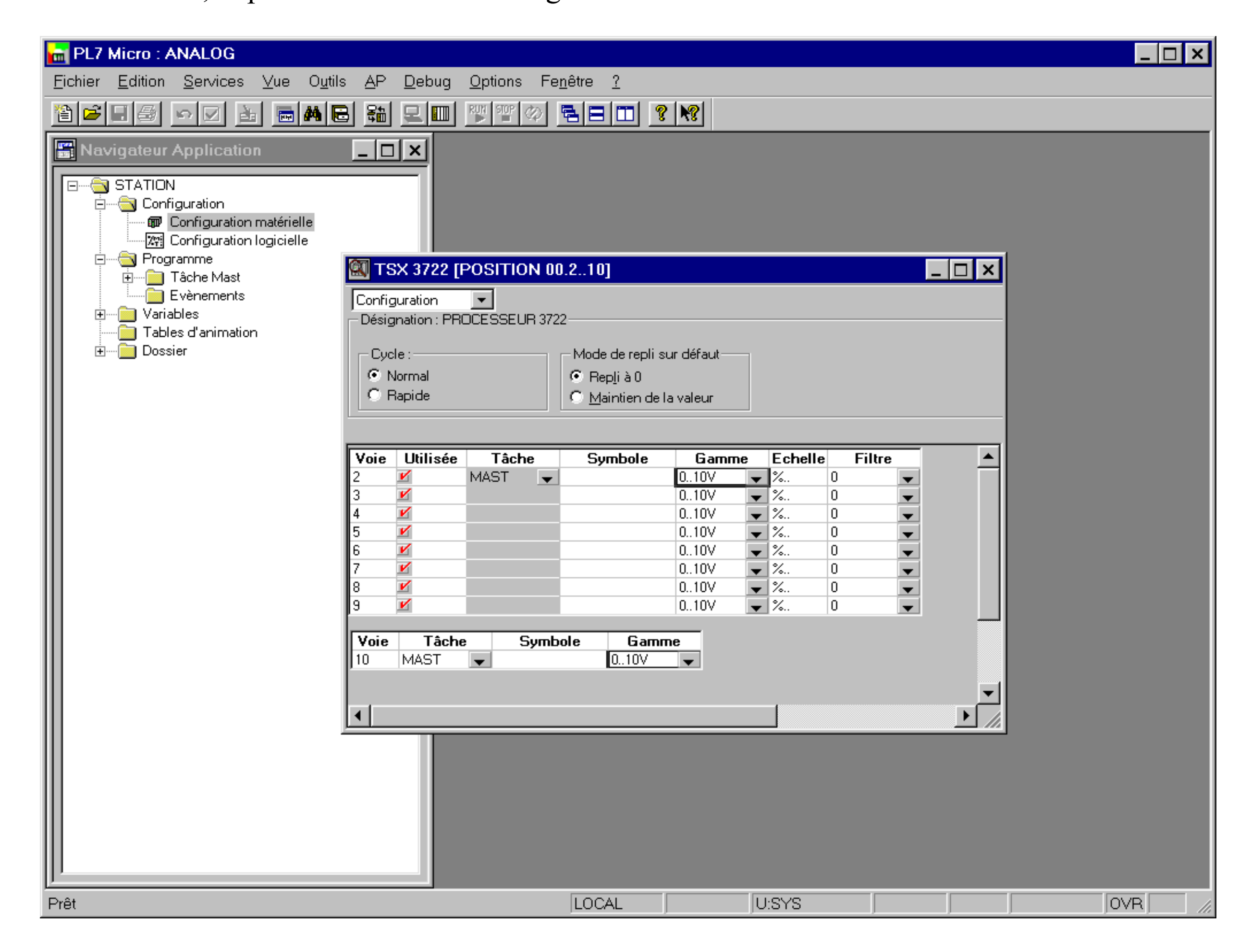


Schéma de câblage

Votre programme devra utiliser une voie d'entrée analogique dans le module intégré à l'automate , la première voie sera configurée en entrée 0-10V.



Réalisez le programme permettant de retrouver dans le mot interne %MW100 la hauteur de liquide (exprimée en cm) correspondant au signal délivré par le potentiomètre .
Le voltmètre connecté sur la sortie analogique va nous permettre de visualiser la tension délivré par le potentiomètre en recopiant l'entrée analogique sur la sortie par un programme .

La relation entre le niveau de la cuve (en cm) et la mesure analogique en (mV) est la suivante :
Y = X a / b + c

Y : la valeur mise à l'échelle (à ranger dans %mw100)
a : l'échelle physique (700-20 = 680)
b : la résolution de l'entrée analogique (10000)
c : l'offset (20)

Utilisez une table d'animation pour visualiser %MW100 .

I.10. Description du TP8 :

1/ Nous vous proposons de réaliser la gestion de deux feux rouge selon le descriptif suivant :

Etape 0 : F1 Rouge , F2 Vert
Etape 1 : F1 Rouge , F2 Orange après 5s
Etape 2 : F1 Rouge , F2 Rouge après 3s
Etape 3 : F1 Vert , F2 Rouge après 2s
Etape 4 : F1 Orange , F2 Rouge après 5s
Etape 5 : F1 Rouge , F2 Rouge après 3s
Etape 0 : F1 Rouge , F2 Vert après 2s

2/ **N**ous voulons donner accès au policier pour passer par un switch **SW1** du mode automatique au mode manuel .

En mode manuel :

- le cycle se bloquera à l'étape où il se trouve .

- le policier passera d'une étape à l'autre par l'activation d'un switch **SW2.**

3/ **L**e policier désir accéder directement aux étapes 0 et 2 par un switch **SW3** .
4/ **L**e policier désir accéder directement a l'étape 3 par un switch **SW4** .

SW1 = %I3.0

SW2 = %I3.1

SW3 = %I3.2

SW4 = %I3.3

F1 Rouge = %Q3.8

F1 Orange = %Q3.9

F1 Vert = %Q3.10

F2 Rouge = %Q4.8

F2 Orange = %Q4.9

F2 Vert = %Q4.10

III - Diagnostic des problèmes de fonctionnement d'un automatisme simple commandé par un automate.

EXERCICE 1 :

QUESTION N° 1 :

- Quel est l'état du bouton RUN, situé sur la face avant de l'automate, quand ce dernier est en mode RUN (Exécution du programme).

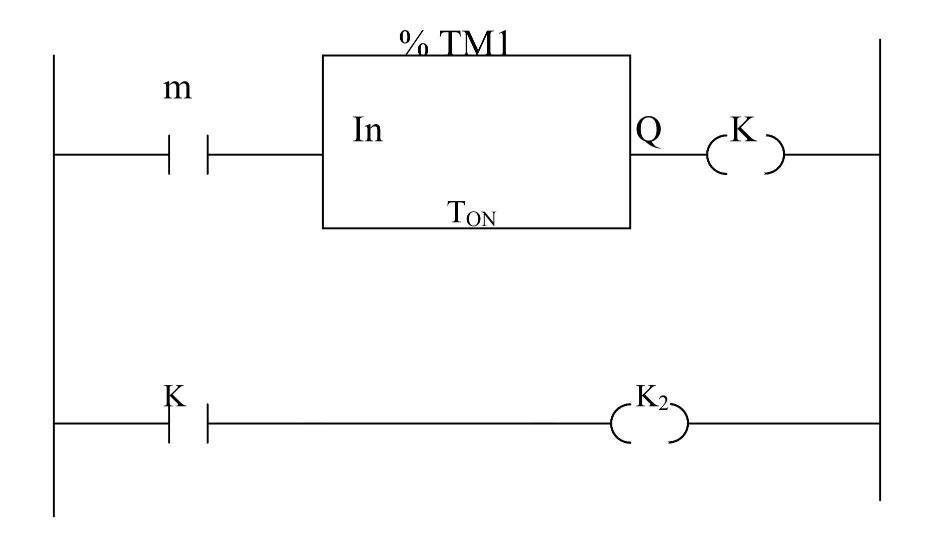
QUESTION N° 2 :

- Compléter le tableau suivant :

Voyant	état	Donner le diagnostic
RUN	fixe	
TER	clignote	
EER	clignote	

EXERCICE 2:

Il s'agit de cet exemple , on constate que la sortie K2 ne fonctionne pas
Analyser le problème en citant toutes les causes possibles pouvant être la source de non fonctionnement de cet automatisme



EXERCICE 3:

1/

1-1/ **R**éaliser un programme en langage ladder pour une commande de marche manuelle d'un Moteur à un seul sens de marche M1 .

Arrêt = %I3.0
Marche = %I3.1
Bobine = %Q3.8

1-2/ **P**rendre en compte la signalisation de l'état du moteur :

- Sur une lampe l'état de marche = %Q3.9
- Sur une autre lampe l'état d'arrêt = %Q3.10

2/

Modifier le programme pour avoir un fonctionnement en deux sens.

IV - L'essai d'un automatisme simple commandé par un automate

EXERCICE 1:

Question n°1 :

Citer certains dangers causés par le forçage des entrées / sorties.

Question n°2 :

Dans le démarrage à double sens de rotation, comment se fait le verrouillage par logiciel et le verrouillage matériel (mécanique) ?

EXERCICE 2 :

- Programmer le démarrage étoile triangle d'un moteur asynchrone.
- Après avoir câblé les entrées et les sorties effectuer l'essai en laissant l'automatisme fonctionner pendant 1 heure.
- Si le système ne fonctionne pas déceler la panne ; changer l'élément défectueux et refaire l'essai.

Evaluation de fin de module :

EXERCICE N° 1

Nous avons une machine à deux moteurs M1 et M2 de telle façon que M2 démarre automatiquement après 15 s de marche du moteur M1 .

Faire un programme automate qui nous permettra de commander le démarrage et l'arrêt des deux moteur à l'aide d'une switch Marche et Arrêt BP1 et BP2.

Noter que :

- Si le moteur M1 s'arrête avant les 15 s et redémarre , on doit réinitialiser la temporisation .
- Si le moteur M1 s'arrête après les 15 s le moteur M2 doit s'arrêter immédiatement .

BP1: %I3.0
BP2: %I3.1
Commande M1 : %Q3.8

Commande M2 : %Q3.9

EXERCICE N° 2

Pour la sécurité de fonctionnement , on installe deux capteurs pour indiquer l'état d'échauffement de chaque Moteur et une porte de sécurité pour le moteur M2 .
Nous voulons regrouper les sécurités de chaque moteur et que si on a un défaut de sécurité pour un des deux moteurs :

- Le défaut de sécurité doit être maintenu jusqu'au disparition de la cause et acquittement .
- Le défaut de sécurité doit arrêter immédiatement le moteur correspondant .
- Le moteur M2 s'arrête soit après un défaut correspondant , soit après un arrêt continue de 3s du moteur M1 .

Sécurité du moteur M1 : %I3.2

Sécurité du moteur M2 : %I3.3

FDC Porte Ouverte M2 : %I3.4

Acquittement Défaut : %I3.5

EXERCICE N° 3

Pour réaliser un entretien périodique du moteur M1 nous voulons que l'automate bloque le démarrage du moteur après 10 démarrages successifs et signale la demande d'entretien .

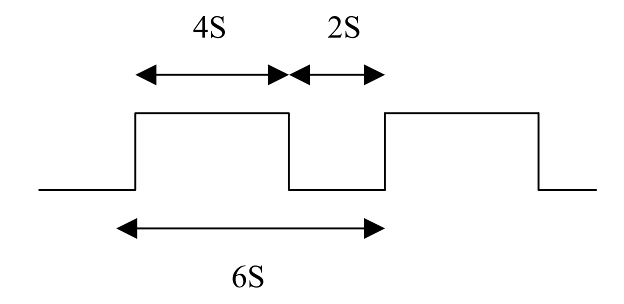
- Le blocage entretien devra être traité de la même manière qu'un défaut sécurité M1 .
- L'opérateur devra avoir accès à initialiser l'entretien et débloquer le démarrage de M1par bouton poussoir BP3 .

BP3 : %I3.6

Demande entretien : %Q3.10

EXERCICE N° 4

L'opérateur demande que le signal de demande d'entretien soit du type clignotement de telle manière qu'il soit 4s allumé et 2s éteint .



Liste bibliographique

- Documentation Télémécanique

- Séminaires A P I